ENC424J600/624J600
Data Sheet

Stand-Alone 10/100 Ethernet Controller with SPI or Parallel Interface

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

ENC424J600/624J600

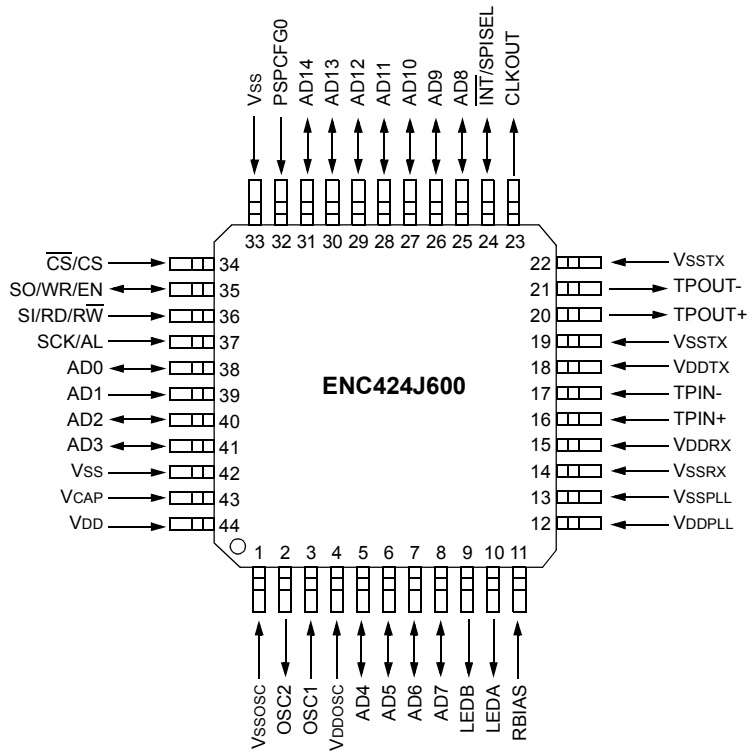
Stand-Alone 10/100 Ethernet Controller with SPI or Parallel Interface

- IEEE 802.3™ Compliant Fast Ethernet Controller
- Integrated MAC and 10/100Base-T PHY
- Hardware Security Acceleration Engines
- 24-Kbyte Transmit/Receive Packet Buffer SRAM
- Supports one 10/100Base-T Port with Automatic Polarity Detection and Correction
- Supports Auto-Negotiation
- Support for Pause Control Frames, including Automatic Transmit and Receive Flow Control
- Supports Half and Full-Duplex Operation
- Programmable Automatic Retransmit on Collision
- Programmable Padding and CRC Generation
- Programmable Automatic Rejection of Erroneous and Runt Packets
- Factory Preprogrammed Unique MAC Address
- MAC:
  - Support for Unicast, Multicast and Broadcast packets
  - Supports promiscuous reception
  - Programmable pattern matching
  - Programmable filtering on multiple packet formats, including Magic Packet™, Unicast, Multicast, Broadcast, specific packet match, destination address hash match or any packet
- PHY:
  - Wave shaping output filter
  - Internal Loopback mode
  - Energy Detect Power-Down mode
- Available MCU Interfaces:
  - 14 Mbit/s SPI interface with enhanced set of opcodes (44-pin and 64-pin packages)
  - 8-bit multiplexed parallel interface (44-pin and 64-pin packages)
  - 8-bit or 16-bit multiplexed or demultiplexed parallel interface (64-pin package only)
- Security Engines:
  - High-performance, modular exponentiation engine with up to 1024-bit operands
  - Supports RSA® and Diffie-Hellman key exchange algorithms
  - High-performance AES encrypt/decrypt engine with 128-bit, 192-bit or 256-bit key
  - Hardware AES ECB, CBC, CFB and OFB mode capability
  - Software AES CTR mode capability
  - Fast MD5 hash computations
  - Fast SHA-1 hash computations
- Buffer:
  - Configurable transmit/receive buffer size
  - Hardware-managed circular receive FIFO
  - 8-bit or 16-bit random and sequential access
  - High-performance internal DMA for fast memory copying
  - High-performance hardware IP checksum calculations
  - Accessible in low-power modes
  - Space can be reserved for general purpose application usage in addition to transmit and receive packets
- Operational:
  - Outputs for two LED indicators with support for single and dual LED configurations
  - Transmit and receive interrupts
  - 25 MHz clock
  - 5V tolerant inputs
  - Clock out pin with programmable frequencies from 50 kHz to 33.3 MHz
  - Operating voltage range of 3.0V to 3.6V
  - Temperature range: -40°C to +85°C industrial
- Available in 44-Pin (TQFP and QFN) and 64-Pin TQFP Packages

Device	SRAM (bytes)	Pin Count	Speed (Mbps)	Security			SPI	PSP			
								Multiplexed		Demultiplexed	
				ModEx 1024-Bit	MD5 SHA-1	AES 256-Bit		8-Bit	16-Bit	8-Bit	16-Bit
ENC424J600	24K	44	10/100	Y	Y	Y	Y	Y	N	N	N
ENC624J600	24K	64	10/100	Y	Y	Y	Y	Y	Y	Y	Y

Pin Diagrams

44-Pin TQFP and QFN



Pin Diagrams (Continued)

64-Pin TQFP

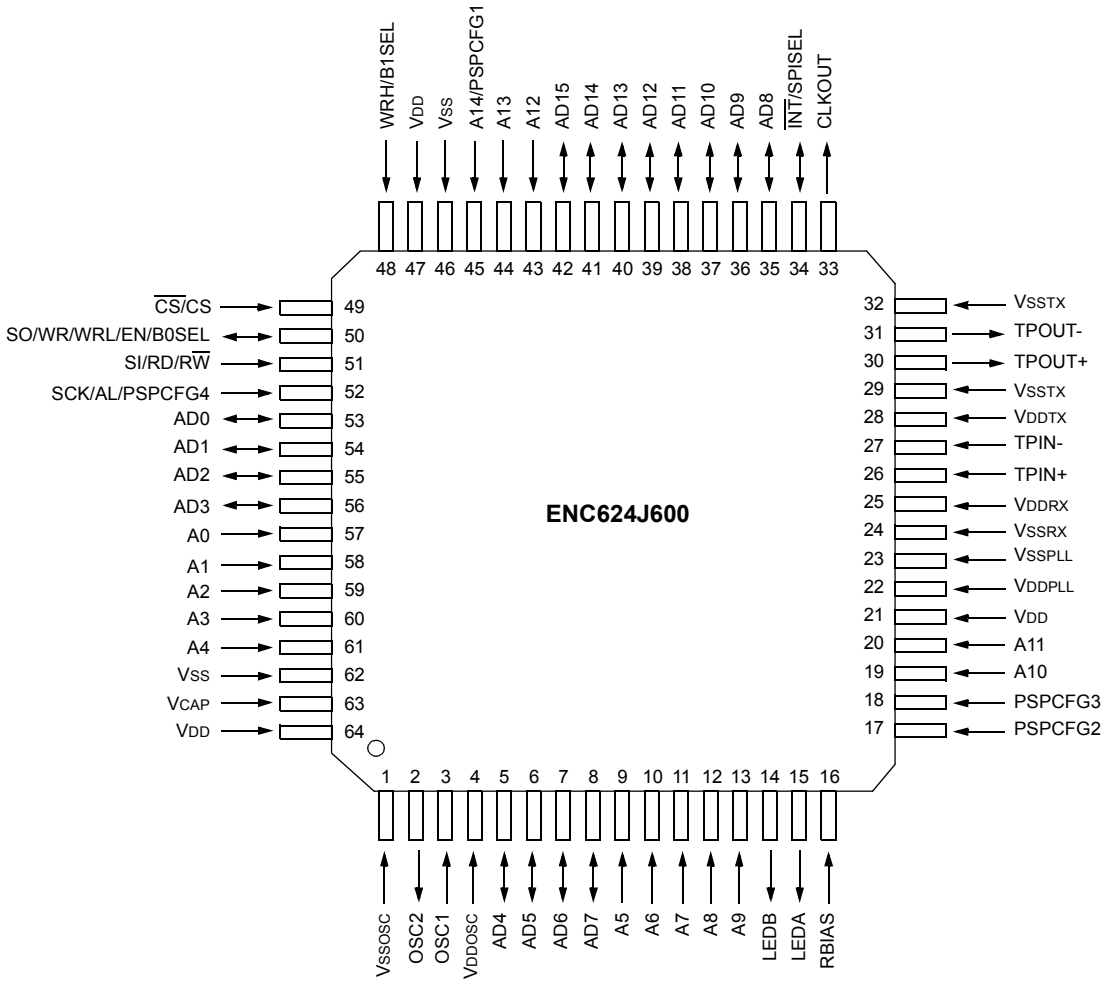


Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

1.0 DEVICE OVERVIEW

This document contains device-specific information for the following devices:

- ENC424J600
- ENC624J600

The ENC424J600 and ENC624J600 are stand-alone, Fast Ethernet controllers with an industry standard Serial Peripheral Interface (SPI) or a flexible parallel interface. They are designed to serve as an Ethernet network interface for any microcontroller equipped with SPI or a standard parallel port.

ENC424J600/624J600 devices meet all of the IEEE 802.3 specifications applicable to 10Base-T and 100Base-TX Ethernet, including many optional clauses, such as auto-negotiation. They incorporate a number of packet filtering schemes to limit incoming packets. They also provide an internal, 16-bit wide DMA for fast data throughput and support for hardware IP checksum calculations.

For applications that require the security and authentication features of SSL, TLS and other protocols related to cryptography, a block of security engines is provided. The engines perform RSA, Diffie-Hellman, AES, MD5 and SHA-1 algorithm computations, allowing reduced code size, faster connection establishment and throughput, and reduced firmware development effort.

Communication with the microcontroller is implemented via the SPI or parallel interface, with data rates ranging from 14 Mbit/s (SPI) to 160 Mbit/s (demultiplexed, 16-bit parallel interface). Dedicated pins are used for LED link and activity indication and for transmit/receive/DMA interrupts.

A generous 24-Kbyte on-chip RAM buffer is available for TX and RX operations. It may also be used by the host microcontroller for general purpose storage. Communication protocols, such as TCP, can use this memory for saving data which may need to be retransmitted.

For easy end product manufacturability, each ENC624J600 family device is preprogrammed with a unique nonvolatile MAC address. In most cases, this allows the end device to avoid a serialized programming step.

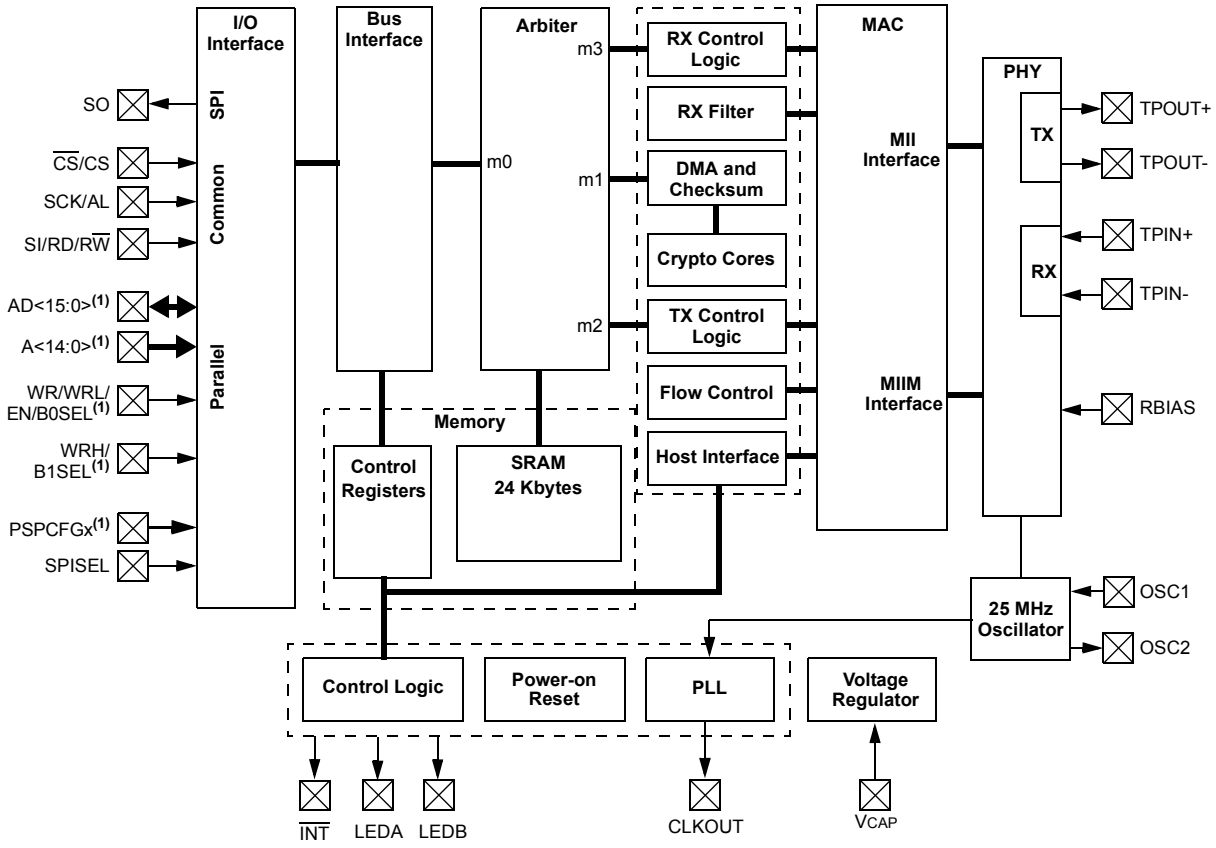
The only functional difference between the ENC424J600 (44-pin) and ENC624J600 (64-pin) devices are the number of parallel interface options they support. These differences, along with a summary of their common features, are provided in Table 1-1. A general block diagram for the devices is shown in Figure 1-1.

A list of the pin features, sorted by function, is presented in Table 1-2.

TABLE 1-1: DEVICE FEATURES FOR ENC424J600/624J600

Feature	ENC424J600	ENC624J600
Pin Count	44	64
Ethernet Operating Speed	10/100 Mbps (auto-negotiate, auto-sense or manual)	
Ethernet Duplex Modes	Half and Full (auto-negotiate and manual)	
Ethernet Flow Control	Pause and Backpressure (auto and manual)	
Buffer Memory (bytes)	24K (organized as 12K word x 16)	
Internal Interrupt Sources	11 (mappable to a single external interrupt flag)	
Serial Host Interface (SPI)	Yes	Yes
Parallel Host Interface:		
Operating modes	2	8
Muliplexed, 8-bit	Yes	Yes
16-bit	No	Yes
Demultiplexed, 8-bit	No	Yes
16-bit	No	Yes
Cryptographic Security Options:		
AES, 128/192/256-bit	Yes	Yes
MD5/SHA-1	Yes	Yes
Modular Exponentiation, 1024-bit	Yes	Yes
Receive Filter Options	Accept or reject packets with CRC match/mismatch, runt error collect or reject, Unicast, Not-Me Unicast, Multicast, Broadcast, Magic Packet™, Pattern Table and Hash Table	
Packages	44-Pin TQFP, QFN	64-Pin TQFP

FIGURE 1-1: ENC424J600/624J600 BLOCK DIAGRAM



Note 1: A<14:0>, AD15, WRL/B0SEL, WRH/B1SEL and PSPCFG<4:1> are available on 64-pin devices only. PSPCFG0 is available on 44-pin devices only.

TABLE 1-2: ENC424J600/624J600 PINOUT DESCRIPTIONS

Pin Name	Pin Number		Pin Type	Input Buffer	Description
	44-Pin	64-Pin			
AD0	38	53	I/O	CMOS	PSP Multiplexed Address Input and/or Bidirectional Data Bus
AD1	39	54	I/O	CMOS	
AD2	40	55	I/O	CMOS	
AD3	41	56	I/O	CMOS	
AD4	5	5	I/O	CMOS	
AD5	6	6	I/O	CMOS	
AD6	7	7	I/O	CMOS	
AD7	8	8	I/O	CMOS	
AD8	25	35	I/O	CMOS	
AD9	26	36	I/O	CMOS	
AD10	27	37	I/O	CMOS	
AD11	28	38	I/O	CMOS	
AD12	29	39	I/O	CMOS	
AD13	30	40	I/O	CMOS	
AD14	31	41	I/O	CMOS	
AD15	—	42	I/O	CMOS	
A0	—	57	I	CMOS	PSP Demultiplexed Address Input Bus
A1	—	58	I	CMOS	
A2	—	59	I	CMOS	
A3	—	60	I	CMOS	
A4	—	61	I	CMOS	
A5	—	9	I	CMOS	
A6	—	10	I	CMOS	
A7	—	11	I	CMOS	
A8	—	12	I	CMOS	
A9	—	13	I	CMOS	
A10	—	19	I	CMOS	
A11	—	20	I	CMOS	
A12	—	43	I	CMOS	
A13	—	44	I	CMOS	
A14	—	45	I	CMOS	
AL	37	52	I	CMOS	PSP Address Latch
B0SEL	—	50	I	CMOS	PSP Byte 0 Select
B1SEL	—	48	I	CMOS	PSP Byte 1 Select
CLKOUT	23	33	O	—	Programmable Clock Output for External Use
$\overline{CS}$	34	49	I	CMOS	SPI Chip Select (active-low)
CS	34	49	I	CMOS	PSP Chip Select (active-high)
EN	35	50	I	CMOS	PSP R/$\overline{W}$ Enable strobe
$\overline{INT}$	24	34	O	—	Interrupt Output (active-low)
LEDA	10	15	O	—	Programmable Ethernet Status/Activity LED
LEDB	9	14	O	—	Programmable Ethernet Status/Activity LED

Legend: I = Input; O = Output; P = Power; CMOS = CMOS compatible input buffer; ANA = Analog level input/output

TABLE 1-2: ENC424J600/624J600 PINOUT DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number		Pin Type	Input Buffer	Description
	44-Pin	64-Pin			
OSC1	3	3	I	ANA	25 MHz Crystal Oscillator/Clock Input
OSC2	2	2	O	—	25 MHz Crystal Oscillator Output
PSPCFG0	32	—	I	CMOS	PSP Mode Select 0
PSPCFG1	—	45	I	CMOS	PSP Mode Select 1
PSPCFG2	—	17	I	CMOS	PSP Mode Select 2
PSPCFG3	—	18	I	CMOS	PSP Mode Select 3
PSPCFG4	—	52	I	CMOS	PSP Mode Select 4
RBIAS	11	16	I	ANA	PHY Bias (external resistor) Connection
RD	36	51	I	CMOS	PSP Read Strobe
R$\overline{W}$	36	51	I	CMOS	PSP Combined Read/$\overline{\text{Write}}$ Signal
SCK	37	52	I	CMOS	SPI Serial Clock Input
SI	36	51	I	CMOS	SPI Serial Data Input (from Master)
SO	35	50	O	—	SPI Serial Data Out (to Master)
SPISEL	24	34	I	CMOS	SPI/PSP Interface Select
TPIN-	17	27	I	ANA	Differential Ethernet Receive Minus Signal Input
TPIN+	16	26	I	ANA	Differential Ethernet Receive Plus Signal Input
TPOUT-	21	31	O	—	Differential Ethernet Transmit Minus Signal Output
TPOUT+	20	30	O	—	Differential Ethernet Transmit Plus Signal Output
V_{CAP}	43	63	P	—	Regulator External Capacitor connection
V_{DD}	44	21, 47, 64	P	—	Positive 3.3V Power Supply for Digital Logic
V_{DDOSC}	4	4	P	—	Positive 3.3V Power Supply for 25 MHz Oscillator
V_{DDPLL}	12	22	P	—	Positive 3.3V Power Supply for PHY PLL Circuitry
V_{DDRX}	15	25	P	—	Positive 3.3V Power Supply for PHY RX Circuitry
V_{DDTX}	18	28	P	—	Positive 3.3V Power Supply for PHY TX Circuitry
V_{SS}	33, 42	46, 62	P	—	Ground Reference for Digital Logic
V_{SSOSC}	1	1	P	—	Ground Reference for 25 MHz Oscillator
V_{SSPLL}	13	23	P	—	Ground Reference for PHY PLL Circuitry
V_{SSRX}	14	24	P	—	Ground Reference for PHY RX Circuitry
V_{SSTX}	19, 22	29, 32	P	—	Ground Reference for PHY TX Circuitry
WR	35	50	I	CMOS	PSP Write Strobe
WRH	—	48	I	CMOS	PSP Write High Strobe
WRL	—	50	I	CMOS	PSP Write Low Strobe

Legend: I = Input; O = Output; P = Power; CMOS = CMOS compatible input buffer; ANA = Analog level input/output

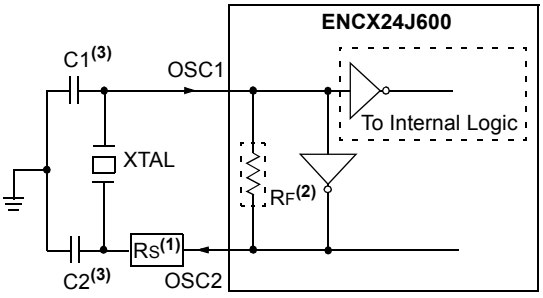
2.0 EXTERNAL CONNECTIONS

2.1 Oscillator

ENC424J600/624J600 devices are designed to operate from a fixed 25 MHz clock input. This clock can be generated by an external CMOS clock oscillator or a parallel resonant, fundamental mode 25 MHz crystal attached to the OSC1 and OSC2 pins. Use of a crystal, rated for series resonant operation, will oscillate at an incorrect frequency. To comply with IEEE 802.3 Ethernet timing requirements, the clock must have no more than ±50 ppm of total error; avoid using resonators or clock generators that exceed this margin.

When clocking the device using a crystal, follow the connections shown in Figure 2-1. When using a CMOS clock oscillator or other external clock source, follow Figure 2-2.

FIGURE 2-1: CRYSTAL OSCILLATOR OPERATION

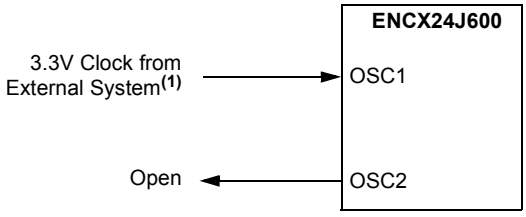


Note 1: A series resistor, RS, may be required for crystals with a low drive strength specification or when using large loading capacitors.

2: The feedback resistor, R_F, is typically 1.5 MΩ (approx).

3: The load capacitors' value should be derived from the capacitive loading specification provided by the crystal manufacture.

FIGURE 2-2: EXTERNAL CLOCK SOURCE



Note 1: Duty cycle restrictions must be observed.

2.2 CLKOUT Pin

The Clock Out pin (CLKOUT) is provided for use as the host controller clock or as a clock source for other devices in the system. Its use is optional.

The 25 MHz clock applied to OSC1 is multiplied by a PLL to internally generate a 100 MHz base clock. This 100 MHz clock is driven through a configurable postscaler to yield a wide range of different CLKOUT frequencies. The PLL multiplication adds clock jitter, subject to the PLL jitter specification in **Section 17.0 "Electrical Characteristics"**. However, the postscaler ensures that the clock will have a nearly ideal duty cycle.

The CLKOUT function is enabled and the postscaler is selected via the COCON<3:0> bits (ECON2<11:8>). To create a clean clock signal, the CLKOUT output and COCON bits are unaffected by all resets and power-down modes. The CLKOUT function is enabled out of POR and defaults to producing a 4 MHz clock. This allows the device to directly clock the host processor.

When the COCON bits are written with a new configuration, the CLKOUT output transitions to the new frequency without producing any glitches. No high or low pulses with a shorter period than the original or new clock are generated.

2.3 Voltage and Bias Pin

2.3.1 VDD AND VSS PINS

To reduce on-die noise levels and provide for the high-current demands of Ethernet, there are many power pins on ENC424J600/624J600 devices:

- VDD and VSS
- VDDOSC and VSSOSC
- VDDPLL and VSSPLL
- VDDRX and VSSRX
- VDDTX and VSSTX

Each VDD and VSS pin pair above should have a 0.1 μF ceramic bypass capacitor placed as close to the pins as possible. For best EMI emission suppression, other smaller capacitors, such as 0.001 μF, should be placed immediately across VDDTX/VSSTX and VDDPLL/VSSPLL.

All VDD power supply pins must be externally connected to the same 3.3V ±10% power source. Similarly, all VSS supply references must be externally connected to the same ground node. If a ground connection appears on two pins (e.g., VSSTX), connect both pins; do not allow either to float. In addition, it is recommended that the exposed bottom metal pad on the 44-pin QFN package be tied to VSS.

Placing ferrite beads or inductors between any two of the supply pins (e.g., between VDDOSC and VDDRX) is not recommended. However, it is acceptable to isolate all of the VDD supplies from the main circuit power supply through a single ferrite bead or inductor, if desired for supply noise suppression reasons. Such isolation is generally not necessary.

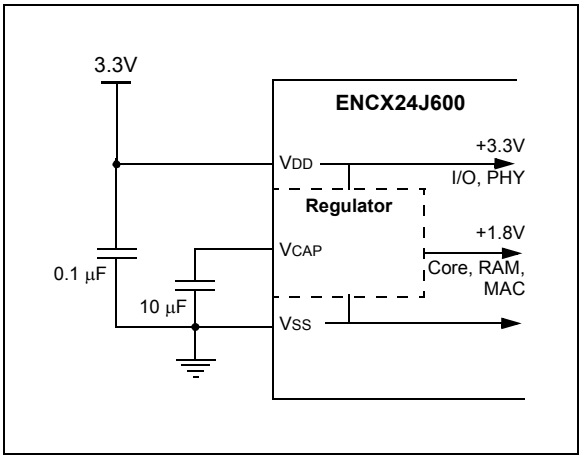
2.3.2 VCAP PIN

Most of the device's digital logic operates at a nominal 1.8V. This voltage is supplied by an on-chip voltage regulator, which generates the digital supply voltage from the VDD rail. The only external component required is an external filter capacitor, connected from the VCAP pin to ground, as shown in Figure 2-3. A value of at least 10 μF is recommended.

The capacitor must also have a relatively low Equivalent Series Resistance (ESR). It is recommended that a low-ESR capacitor (ceramic, tantalum or similar) should be used and high-ESR capacitors (such as aluminum electrolytic) should be avoided.

The internal regulator is not designed to drive external loads; therefore, do not attach other circuitry to VCAP.

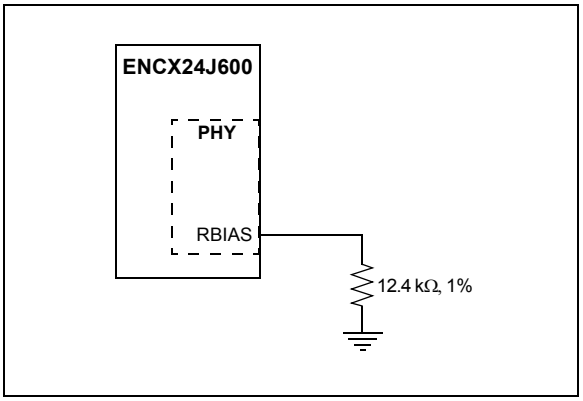
FIGURE 2-3: VCAP CONNECTIONS



2.3.3 RBIAS PIN

The internal analog circuitry in the PHY module requires that an external 12.4 kΩ, 1% resistor be attached from RBIAS to ground, as shown in Figure 2-4. The resistor influences the TPOUT+/- signal amplitude. The RBIAS resistor should be placed as close as possible to the chip with no immediately adjacent signal traces in order to prevent noise capacitively coupling into the pin and affecting the transmit behavior. It is recommended that the resistor be a surface mount type.

FIGURE 2-4: RBIAS RESISTOR



2.4 Ethernet Signal Pins and External Magnetics

Typical applications for ENC424J600/624J600 devices require an Ethernet transformer module, and a few resistors and capacitors to implement a complete IEEE 802.3 compliant 10/100 Ethernet interface, as shown in Figure 2-5.

The Ethernet transmit interface consists of two pins: TPOUT+ and TPOUT-. These pins implement a differential pair and a current-mode transmitter. To generate an Ethernet waveform, ordinary applications require the use of a 1:1 center tapped pulse transformer, rated for 10/100 or 10/100/1000 Ethernet operations. When the Ethernet module is enabled and linked with a partner, current is continually sunk through both TPOUT pins. When the PHY is actively transmitting, a differential voltage is created on the Ethernet cable by varying the relative current sunk by TPOUT+ compared to TPOUT-.

The Ethernet receive interface similarly consists of a differential pair: TPIN+ and TPIN-. To meet IEEE 802.3 compliance and help protect against electrostatic discharge, these pins are normally isolated from the Ethernet cable by a 1:1 center tapped transformer (available in the same package as the TX transformer).

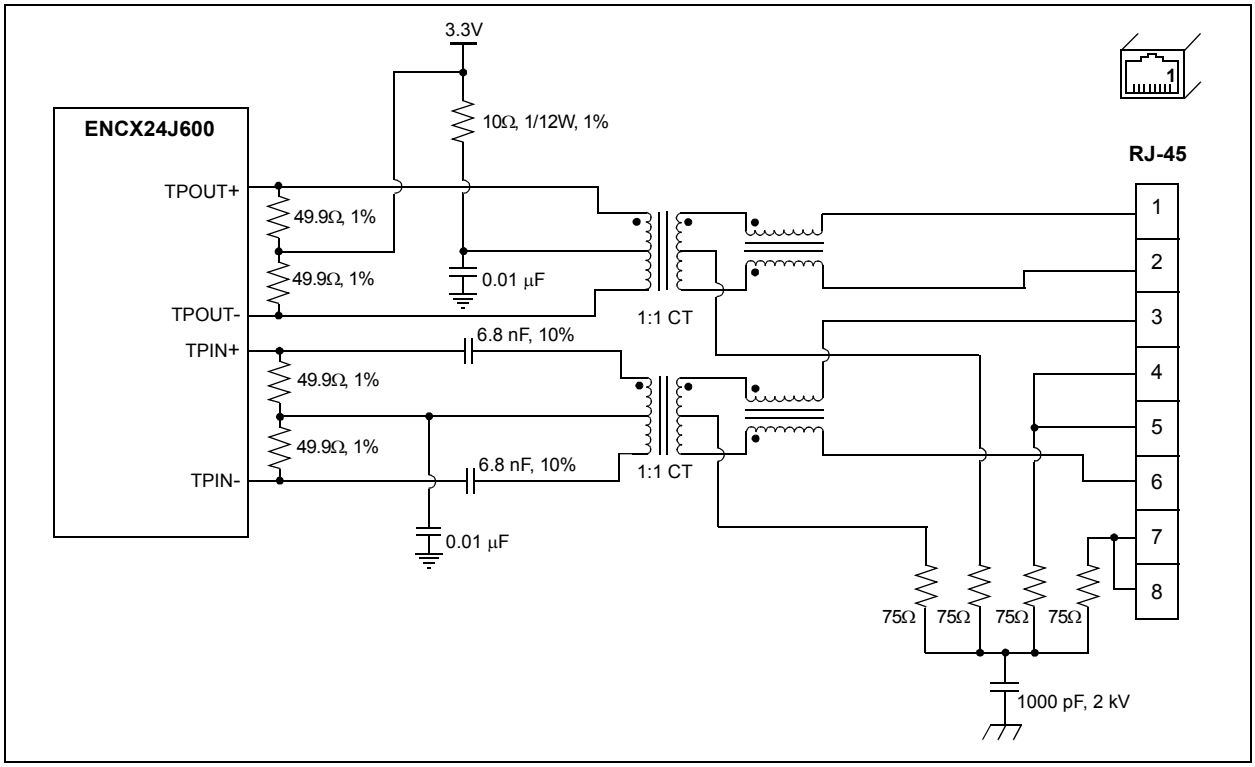
Internally, the PHY uses a high-speed ADC to sample the receive waveform and decodes it using a DSP. The PHY implements many robustness features, including baseline wander correction (applicable to 100Base-TX) and automatic RX polarity correction (applicable to 10Base-T).

Four 49.9Ω, 1% resistors are required for proper termination of the TX and RX transmission lines. If the board layout necessitates long traces between the ENCX24J600 and Ethernet transformers, the termination resistors should be placed next to the silicon instead of the transformers.

On the receive signal path, two 6.8 nF 10% capacitors are used. These capacitors, in combination with the 49.9Ω termination resistors, form an RC high-pass filter to reduce baseline wander. For best performance, these capacitors should not be omitted or changed. The various remaining capacitors provide DC current blocking and provide stability to the common-mode voltage of both of the differential pairs. The TPIN+/- pins weakly output a common-mode voltage that is acceptable to the internal ADC. For proper operation, do not attempt to externally force the TPIN+/- common-mode voltage to some other value.

The 10Ω 1% resistor provides a current path from the power supply to the center tap of the TX transformer. As mentioned previously, the TPOUT+/- pins implement a Current mode drive topology in which the pins are only capable of sinking current; they do not produce a direct voltage. This current path through the transformer generates the transmit waveform. The 10Ω resistor reduces the amount of heat that the PHY would have to dissipate, and therefore, must have a power rating of 1/12W or better.

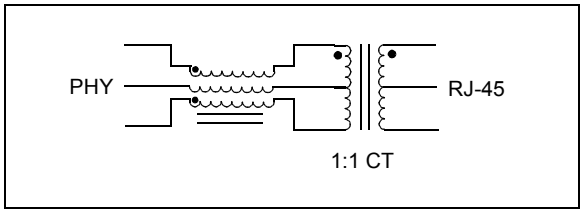
FIGURE 2-5: TYPICAL ETHERNET MAGNETICS CONNECTIONS



2.4.1 ADDITIONAL EMI AND LAYOUT CONSIDERATIONS

To reduce EMI emissions, common-mode chokes are shown adjacent to the transformers on the cable (RJ-45) side. These chokes come standard in typical Ethernet transformer modules. Because the ENCX24J600 PHY uses a current-mode drive topology, the transmit choke must normally be located on the cable side of the transmit transformer. Orienting the magnetics such that the choke is on the PHY side of the transmit transformer usually results in a distorted, non-compliant transmit waveform. However, some magnetics which wrap the TX center tap wire around the TX choke core can also be used to generate a compliant waveform (Figure 2-6). These types of transformers may be desirable in some Power-over Ethernet (PoE) applications.

FIGURE 2-6: ALTERNATE TX CHOKE TOPOLOGY



The common-mode choke on the RX interface can be placed on either the cable side or PHY side of the receive transformer. Recommended and required magnetics characteristics are located in **Section 17.0 "Electrical Characteristics"**.

The four 75Ω resistors and high-voltage capacitor in Figure 2-5 are intended to prevent each of the twisted pairs in the Ethernet cables from floating and radiating EMI. Their implementation may require adjustment in PoE applications.

Unless the TX and RX signal pairs are kept short, they should be routed between the ENCX24J600 and the Ethernet connector following differential routing rules. Like Ethernet cables, 100Ω characteristic impedance should be targeted for the differential traces. The use of vias, which introduce impedance discontinuities, should be minimized. Other board level signals should not run immediately parallel to the TX and RX pairs to minimize capacitive coupling and crosstalk.

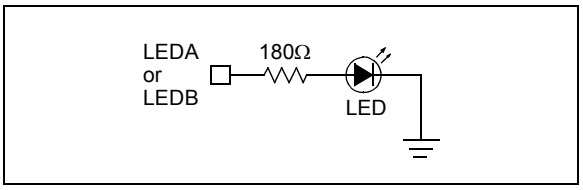
2.5 LEDA and LEDB Pins

The LEDA and LEDB pins provide dedicated LED status indicator outputs. The LEDs are intended to display link status and TX/RX activity among other programmable options; however, the use of one or both is entirely optional. The pins are driven automatically by the hardware and require no support from the host microcontroller. Aside from the LEDs themselves, a current-limiting resistor is generally the only required component.

By default on POR, LEDA displays the Ethernet link status, while LEDB displays PHY-level TX/RX activity. Because the LEDs operate at the PHY level, RX activity will be displayed on LEDB any time Ethernet packets are detected, regardless of if the packet is valid and meets the correct RX filtering criteria.

Normally, the device illuminates the LED by sourcing current out of the pin, as shown in Figure 2-7. Connecting the LED in reverse, with the anode connected to V_{DD} and the cathode to LEDA/LEDB (through a current-limiting resistor), causes the LED to show "inverted sense" behavior, lighting the LED when it should be off and extinguishing the LED when the LED should be on.

FIGURE 2-7: SINGLE COLOR LED CONNECTION

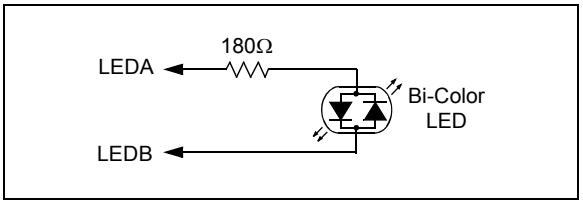


Both LEDs automatically begin operation whenever power is applied, a 25 MHz clock is present and the Ethernet magnetics are present and wired correctly. A connection to the host microcontroller via the SPI or PSP interface is not required. LEDA and LEDB can, therefore, be used as a quick indicator of successful assembly during initial prototype development.

2.5.1 USING BI-COLOR LEDs

In space constrained applications, it is frequently desirable to use a single bi-color LED to display multiple operating parameters. These LEDs are connected between LEDA and LEDB, as shown in Figure 2-8.

FIGURE 2-8: BI-COLOR LED CONNECTION



ENCX24J600 devices include two special hardware display modes to make maximal use of a bi-color LED. These modes are selected when the LACFG<3:0> and LBCFG<3:0> bits (EIDLED<15:8>) are set to '1111' or '1110'. In these configurations, the link state turns the LED on, the speed/duplex state sets the LED color and TX/RX events cause the LED to blink off. If a link is present, no TX/RX events are occurring and the speed/duplex state is 100 Mbps/full duplex, respectively, then the LEDB pin will be driven high while LEDA will be driven low.

2.6 $\overline{\text{INT}}$ Pin

The $\overline{\text{INT}}$ pin is an active-low signal that is used to flag interrupt events to external devices. Depending on the application, it can be used to signal the host microcontroller whenever a packet has been received or transmitted, or that some other asynchronous operation has occurred. It can also be used to wake-up the microcontroller or other system components based on LAN activity; its use is optional.

The $\overline{\text{INT}}$ pin is driven high when no interrupt is pending and is driven low when an interrupt has occurred. It does not go into a high-impedance state, except during initial power-on while the multiplexed SPISEL pin function is being used.

Since ENC424J600/624J600 devices incorporate a buffer for storing transmit and receive packets, the host microcontroller never needs to perform real-time operations on the device. The microcontroller can poll the device registers to discover if the device status has changed.

2.7 Host Interface Pins

For the maximum degree of flexibility in interfacing with microcontrollers, ENC424J600/624J600 devices offer a choice between a serial interface based on the Serial Peripheral Interface (SPI) standard, and a flexible 8 or 16-bit parallel slave port (PSP) interface. Only one interface may be used at any given time.

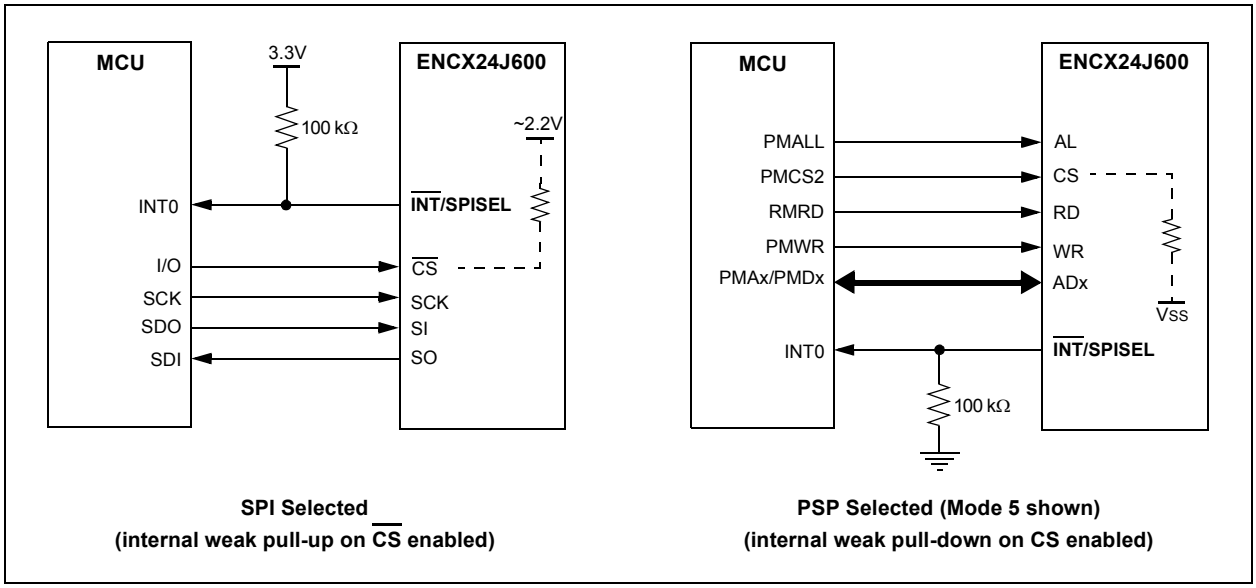
The I/O interface is hardware selected on power-up using the SPISEL function on the $\overline{\text{INT}}$/SPISEL pin. This is done by latching in the voltage level applied to the pin approximately 1 to 10 μs after power is applied to the device and the device exits Power-on Reset. If SPISEL is latched at a logic high state, the serial interface is enabled. If SPISEL is latched at a logic low state, the PSP interface is enabled. Figure 2-9 shows example connections required to select the SPI or PSP interface upon power-up.

To ensure the SPI interface is selected upon power-up, an external pull-up resistor to VDD must be connected to the SPISEL pin. Alternatively, if the parallel interface is to be used, a pull-down resistor to VSS must be connected to the SPISEL pin. In most circuits, it is recommended that a 100 kΩ or smaller resistor be used to ensure that the correct logic level is latched in reliably. If a large capacitance is present in the SPISEL circuit, such as from stray capacitance, a smaller pull-up or pull-down resistor may be required to compensate and ensure the correct level is sensed during power-up.

As SPISEL is multiplexed with the $\overline{\text{INT}}$ interrupt output function, a direct connection to VDD or VSS without a resistor is prohibited. If $\overline{\text{INT}}$ is connected to the host microcontroller, the microcontroller must leave this signal in a high-impedance state and not attempt to drive it to an incorrect logic state during power-up.

If the VDD supply has a slow ramp rate, the device will exit POR, exceed the 1 to 10 μs latch timer and sample the SPISEL pin state before VDD has reached the specified minimum operating voltage of the device. In this case, the device will still latch in the correct value, assuming the minimum VIH (D004) or maximum VIL (D006) specification is met, which is a function of VDD.

FIGURE 2-9: USING THE $\overline{\text{INT}}$/SPISEL PIN TO SELECT THE I/O INTERFACE



2.7.1 SPI

When enabled, the SPI interface is implemented with four pins:

- $\overline{CS}$
- SO
- SI
- SCK

All four of these pins must be connected to use the SPI interface.

The $\overline{CS}$, SI and SCK input pins are 5V tolerant. The SO pin is also 5V tolerant when in a high-impedance state. SO is always high-impedance when $\overline{CS}$ is connected to logic high (i.e., chip not selected).

When the SPI interface is enabled, all PSP interface pins (except PSPCFG2 and PSPCFG3 on ENC624J600 devices) are unused. They are placed in a high-impedance state and their input buffers are disabled. For best ESD performance, it is recommended that the unused PSP pins be tied to either VSS or VDD. However, these pins may be left floating if it is desirable for board level layout and routing reasons.

When using an ENC624J600 device in SPI mode, it is recommended that the PSPCFG2 and PSPCFG3 pins be tied to either VSS or any logic high voltage, and not be left floating. The particular state used is unimportant.

2.7.2 PSP

Depending on the particular device, the PSP interface is implemented with up to 34 pins. The interface is highly configurable to accommodate many different parallel interfaces; not all available pins are used in every configuration. Up to 8 different operating modes are available. These are explained in detail in **Section 5.0 "Parallel Slave Port Interface (PSP)"**.

The PSPCFG pins control which parallel interface mode is used. The values on these pins are latched upon device power-up in the same manner as the SPISEL pin. The combinations of VDD and VSS voltages on the different PSPCFG mode pins determine the PSP mode according to Table 2-1.

On ENC424J600 devices, only PSP Modes 5 and 6 (8-bit width, multiplexed data and address) are available. The mode is selected by applying VSS or VDD, respectively, to PSPCFG0.

On ENC624J600 devices, all eight PSP modes are available and are selected by connecting the PSPCFG<4:1> pins directly to VDD or ground. The mode selection is encoded such that the multiplexed pin functions, AD14 (on PSPCFG1) and SCK/AL (on PSPCFG4), are used only in the "don't care" positions. Therefore, pull-up/pull-down resistors are not required for these pins.

All PSP pins, except for AD<15:0>, are inputs to the ENC624J600 family device and are 5V tolerant. The AD<15:0> pins are bidirectional I/Os and are 5V tolerant in Input mode. The pins are always inputs when the CS signal is low (chip not selected).

Any unused PSP pins are placed in a high-impedance state. However, it is recommended that they be tied to either Vss or a logic high voltage and not be left floating.

TABLE 2-1: PSP MODE SELECTION FOR ENC424J600/624J600 DEVICES

Interface Mode	$\overline{INT}$/SPISEL	PSPCFG					Pins Used
		0	1	2	3	4	
44-Pin							
PSP Mode 5	Pull Down	0	—	—	—	—	AL, CS, RD, WR, AD<14:0>
PSP Mode 6	Pull Down	1	—	—	—	—	AL, CS, R$\overline{W}$, EN, AD<14:0>
64-Pin							
PSP Mode 1	Pull Down	–	x	0	0	0	CS, RD, WR, A14:A0, AD<7:0>
PSP Mode 2	Pull Down	–	x	0	0	1	CS, R$\overline{W}$, EN, A14:A0, AD<7:0>
PSP Mode 3	Pull Down	–	x	1	0	0	CS, RD, WRL, WRH, A<13:0>, AD<15:0>
PSP Mode 4	Pull Down	–	x	1	0	1	CS, R$\overline{W}$, B0SEL, B1SEL, A<13:0>, AD<15:0>
PSP Mode 5	Pull Down	–	0	0	1	x	AL, CS, RD, WR, AD<14:0>
PSP Mode 6	Pull Down	–	1	0	1	x	AL, CS, R$\overline{W}$, EN, AD<14:0>
PSP Mode 9	Pull Down	–	0	1	1	x	AL, CS, RD, WRL, WRH, AD<15:0>
PSP Mode 10	Pull Down	–	1	1	1	x	AL, CS, R$\overline{W}$, B0SEL, B1SEL, AD<15:0>

Legend: x = don't care, 0 = logic low (tied to VSS), 1 = logic high (tied to VDD), — = pin not present

2.7.3 $\overline{CS}$/CS PIN

The chip select functions for the serial and parallel interfaces are shared on one common pin, $\overline{CS}$/CS. This pin is equipped with both internal weak pull-up and weak pull-down resistors. If the SPI interface is selected ($\overline{CS}$), the pull-up resistor is automatically enabled and the pull-down resistor is disabled. If the PSP interface is chosen (CS), the pull-down resistor is automatically enabled and the pull-up resistor is disabled. This allows the $\overline{CS}$/CS pin to stay in the unselected state when not being driven, avoiding the need for an external board level resistor on this pin.

When enabled by using SPI mode, the internal weak pull-up only pulls the $\overline{CS}$/CS pin up to approximately V_{DD}-1.1V or around 2.2V at typical conditions without any loading; it does not pull all the way to V_{DD}. When using the PSP interface, the pull-down will be enabled, which is capable of pulling all the way to V_{SS} when unloaded.

2.8 Digital I/O Levels

All digital output pins on ENC424J600/624J600 devices contain CMOS output drivers that are capable of sinking and sourcing up to 18 mA continuously. All digital inputs and I/O pins operating as inputs are 5V tolerant. These features generally mean that the ENCX24J600 can connect directly to the host microcontroller without the need of any glue logic. However, some consideration may be necessary when interfacing with 5V systems.

Since the digital outputs drive only up to the V_{DD} voltage (3.3V nominally), the voltage may not be high enough to ensure a logical high is detected by 5V systems which have high input thresholds. In such cases, unidirectional level translation from the 3.3V ENCX24J600 up to the 5V host microcontroller may be needed.

When using the SPI interface, an economical 74HCT08 (quad AND gate), 74ACT125 (quad 3-state buffer) or other 5V CMOS chip with TTL level input buffers may be used to provide the necessary level shifting. The use of 3-state buffers permits easy integration into systems which share the SPI bus with other devices. However, users must make certain that the propagation delay of the level translator does not reduce the maximum SPI frequency below desired levels. Figure 2-10 and Figure 2-11 show two example translation schemes.

When using the PSP interface, eight, or all sixteen of the ADx pins, may need level translation when performing read operations on the ENCX24J600. The 8-bit 74ACT245 or 16-bit 74ACT16245 bus transceiver, or similar devices, may be useful in these situations.

FIGURE 2-10: LEVEL SHIFTING ON THE SPI INTERFACE USING AND GATES

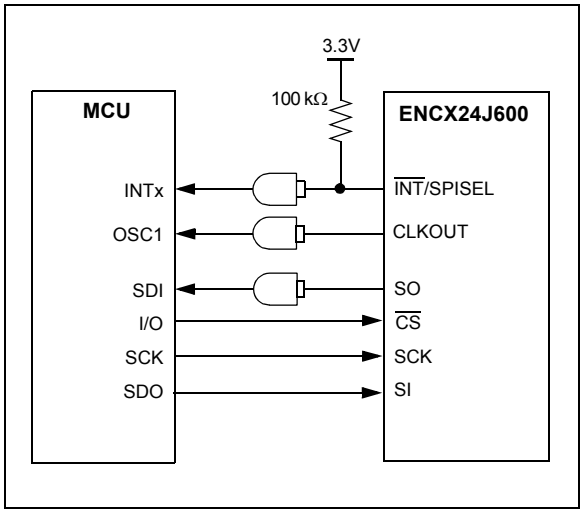
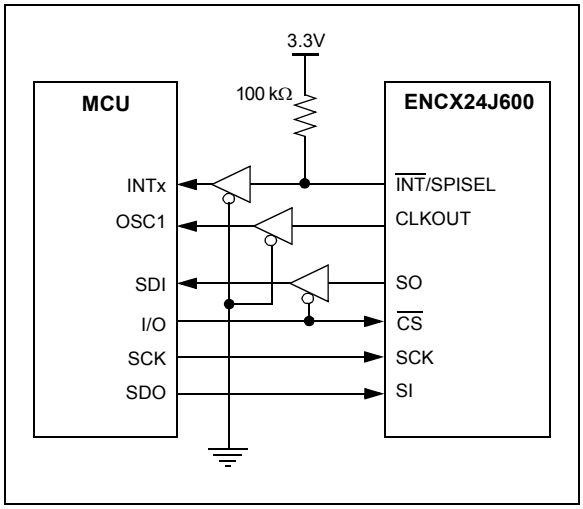


FIGURE 2-11: LEVEL SHIFTING ON THE SPI INTERFACE USING 3-STATE BUFFERS



NOTES:

3.0 MEMORY ORGANIZATION

All memory in ENC424J600/624J600 devices is implemented as volatile RAM. Functionally, there are four unique memories:

- Special Function Registers (SFRs)
- PHY Special Function Registers
- Cryptographic Data Memory
- SRAM Buffer

The SFRs configure, control and provide status information for most of the device. They are directly accessible through the I/O interface.

The PHY SFRs configure, control and provide status information for the PHY module. They are located inside the PHY module and isolated from all other normal SFRs, so they are not directly accessible through the I/O interface.

The cryptography data memory is used to store key and data material for the modular exponentiation, AES and MD5/SHA-1 hashing engines. This memory area can only be accessed through the DMA module.

The SRAM buffer is a bulk 12K x 16-bit (24 Kbyte) RAM array used for TX and RX packet buffering, as well as general purpose storage by the host microcontroller. Although the SRAM uses a 16-bit word, it is byte-writable. This memory is indirectly accessible through pointers on all I/O interfaces. It can also be accessed directly through the PSP interfaces.

3.1 I/O Interface and Memory Map

Depending on the I/O interface selected, the four memories are arranged into two or three different memory address spaces. When the serial interface is selected, the memories are grouped into three address spaces. When one of the parallel interfaces is selected, they are arranged into two address spaces. In all cases, the PHY SFRs reside in their own memory address space.

3.1.1 SPI INTERFACE MAP

When the SPI interface is selected, the device memory map is comprised of three memory address spaces (Figure):

- the SFR area
- the main memory area
- the PHY register area

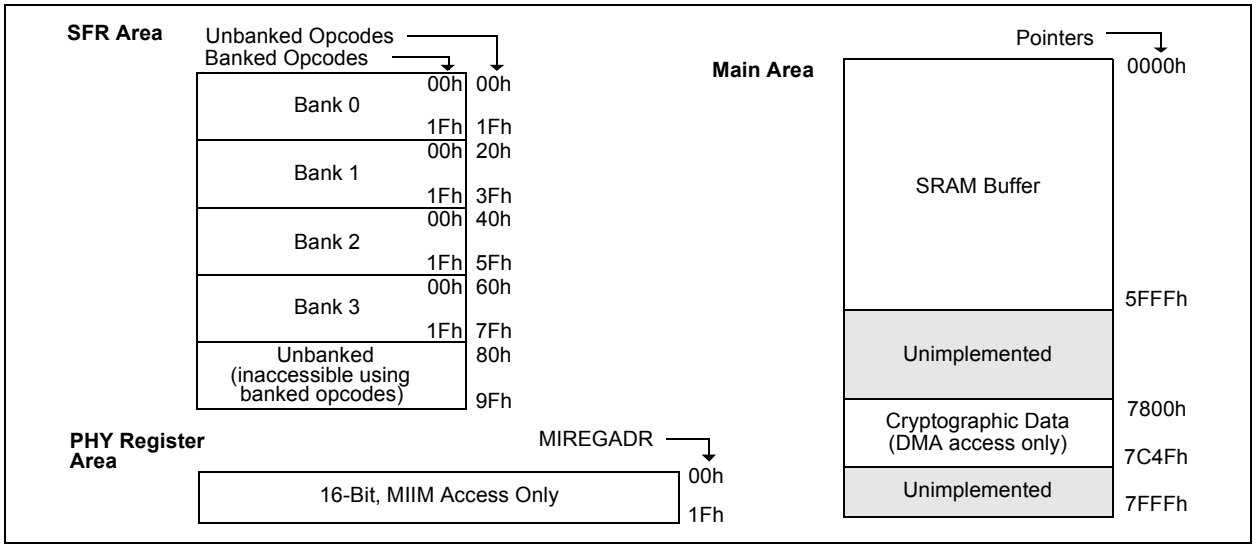
The SFR area is directly accessible to the user. This is a linear memory space that is 160 bytes long. For efficiency, the SFR area can be addressed as four banks of 32 bytes each, starting at the beginning of the space (00h), with an additional unbanked area of 32 bytes at the end of the SFR memory. Banked addressing allows SFRs to be addressed with fewer address bits being exchanged over the serial interface for each transaction. This decreases protocol overhead and enhances performance. SFRs can also be directly addressed by their 8-bit unbanked addresses using unbanked SPI commands. This allows for a simpler interface whenever transaction overhead is not critical.

The main memory area is organized as a linear, byte-addressable space of 32 Kbytes. Of this, the first 24-Kbyte area (0000h through 5FFFh) is implemented as the SRAM buffer. The buffer is accessed by the device using several SFRs as memory pointers and virtual data window registers, as described in **Section 3.5.5 “Indirect SRAM Buffer Access”**.

Addresses in the main memory area, between 7800h and 7C4Fh, are mapped to the memory for the cryptographic data modules. These addresses are not directly accessible through the SPI interface; they can only be accessed through the DMA.

The PHY SFRs are the final memory space. This is a linear, word-addressable memory space of 32 words. This area is only accessible by the MIIM interface (see **Section 3.3 “PHY Special Function Registers”** for more details).

FIGURE 3-1: ENC424J600/624J600 MEMORY MAP WITH SPI INTERFACE



3.1.2 PSP INTERFACE MAPS

When one of the parallel interfaces is selected, the memory map is very different from the SPI map. There are two different memory address spaces (Figure 3-2):

- the main memory area
- the PHY register area

As in the serial memory map, the main memory area is a linear, byte-addressable space of 32 Kbytes, with the SRAM buffer located in the first 24-Kbyte region. The cryptographic data memory is also mapped to the same location as in the serial memory map. The main difference is that the SFRs are now located to an area with a higher address than the cryptographic data space. Additional memory areas above the SFRs are reserved for their accompanying Bit Set and Bit Clear registers.

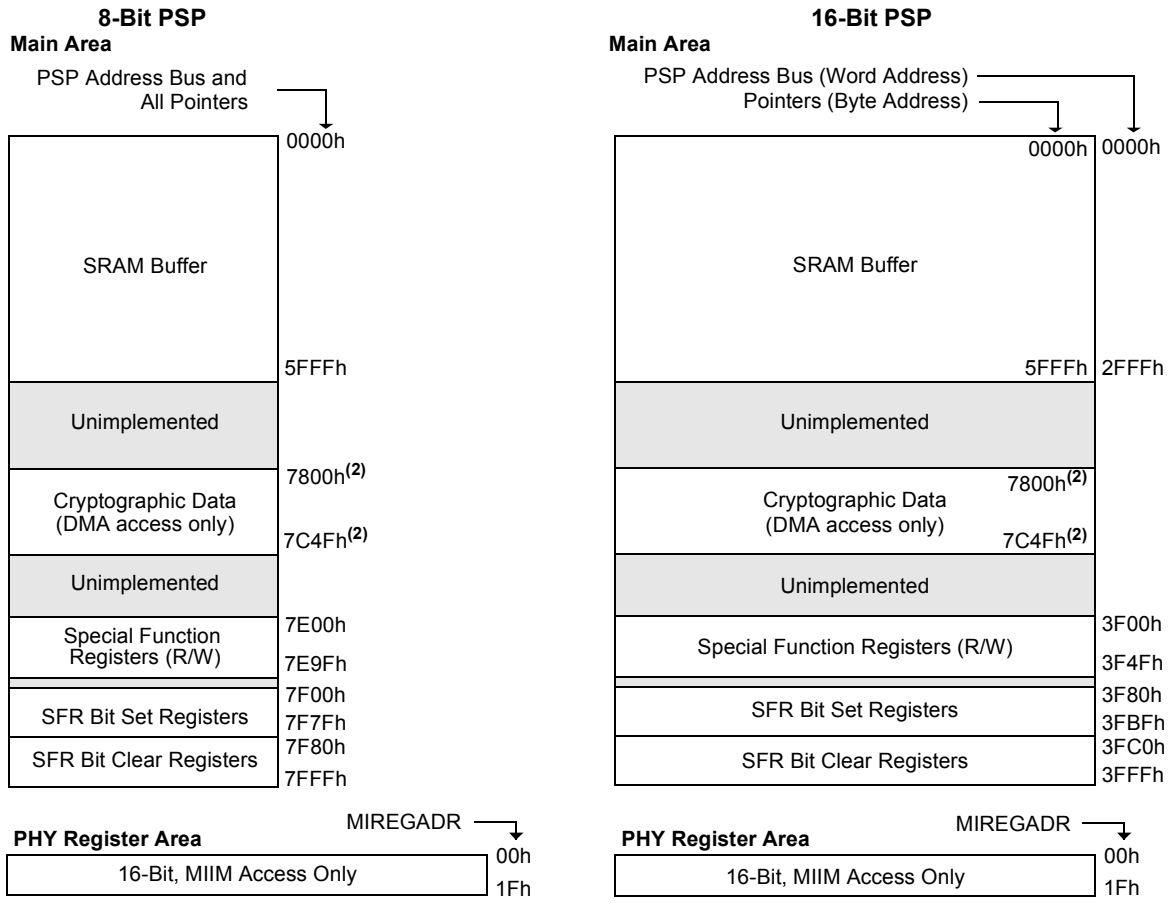
Except for the cryptographic data memory, all addresses in the main memory area are directly accessible using the PSP bus. As with the serial interface, the cryptographic memory can only be accessed through the DMA.

The difference between the 8-bit and 16-bit interfaces is how the SRAM buffer is addressed by the external address bus. In 16-bit data modes, the address bus treats the buffer as a 16-byte wide, word-addressable space, spanning 000h to 3FFFh. In 8-bit data modes, the address bus treats the buffer as an 8-bit, byte-addressable space, ranging from 0000h to 7FFFh. In either case, the SFRs used as memory pointers still address the buffer as a byte-wide, byte-addressable space.

The PHY SFR space is implemented in the same manner as the SPI interface described above.

In both 8-bit and 16-bit PSP modes, full device functionality can be realized without using the full width of the address bus. This is because the SRAM buffer can still be read and written to by using SFR pointers. In practical terms, this can allow designers in space or pin constrained applications to only connect a subset of the A or AD address pins to the host microcontroller. For example, in the 8-Bit Multiplexed PSP Modes 5 or 6, tying pins, AD<14:9> to VDD, still allows direct address access to all SFRs. This reduces the number of pins required for connection to the host controller, including the interface control pins to 12 or 13.

FIGURE 3-2: ENC424J600/624J600 MEMORY MAPS FOR PSP INTERFACES(1)



Note 1: Memory areas not shown to scale.
2: Addresses in this range are accessible only through internal address pointers of the DMA module.

3.2 Special Function Registers

The SFRs provide the main interface between the host controller and the on-chip Ethernet controller logic. Writing to these registers controls the operation of the interface, while reading the registers allows the host controller to monitor operations.

All registers are 16 bits wide. On the SPI and 8-bit PSP interfaces, which are inherently byte-oriented, the registers are split into separate high and low locations which are designated by an “H” or “L” suffix, respectively. All registers are organized in little-endian format such that the low byte is always at the lower memory address.

Some of the available addresses are unimplemented or marked as reserved. These locations should not be written to. Data read from reserved locations should be ignored. Reading from unimplemented locations will return ‘0’. When reading and writing to registers which contain reserved bits, any rules stated in the register definition should be observed.

The addresses of all user-accessible registers are provided in Tables 3-1 through 3-6. A complete bit level listing of the SFRs is presented in Table 3-7 (page 26).

3.2.1 E REGISTERS

SFRs with names starting with “E” are the primary control and pointer registers. They configure and control all of the (non-MAC) top-level features of the device, as well as manipulate the pointers that define the memory buffers. These registers can be read and written in any order, with any length, without concern for address alignment.

3.2.2 MAC REGISTERS

SFRs with names that start with “MA” or “MI” are implemented in the MAC module hardware. For this reason, their operation differs from “E” registers in two ways.

First, MAC registers support read and write operations only. Individual bit set and bit clear operations cannot be performed.

Additionally, MAC registers must always be written as a 16-bit word, regardless of the I/O interface being used. That is, on the SPI or 8-bit PSP interfaces, all write operations must be performed by writing to the low byte, followed by a write to the associated high byte. On 16-bit PSP interfaces, both write enables or byte selects must be asserted to perform the 16-bit write. Non-sequential writes, such as writing to the low byte of one MAC register, the low byte of a second MAC register and then the high byte of the first register cannot be performed.

3.2.3 SPI REGISTER MAP

As previously described, the SFR memory is partitioned into four banks plus a special region that is not bank addressable. Each bank is 32 bytes long and addressed by a 5-bit address value. All SFR memory may also be accessed via unbanked SPI opcodes which use a full 8-bit address to form a linear address map without banking.

The last 10 bytes (16h to 1Fh) of all SPI banks point to a common set of five registers: EUDAST, EUDAND, ESTAT, EIR and ECON1. These are key registers used in controlling and monitoring the operation of the device. Their common banked addresses allow easy access without switching the bank.

The SPI interface implements a comprehensive instruction set that allows for reading and writing of registers, as well as setting and clearing individual bits or bit fields within registers. The SPI instruction set is explained in detail in **Section 4.0 “Serial Peripheral Interface (SPI)”**.

The SFR map for the SPI interface is shown in Table 3-1. Registers are presented by a bank. The banked (5-bit) address applicable to the registers in each row is shown in the left most column. The unbanked (8-bit) address for each register is shown to the immediate left of the register name.

Note:	SFRs in the unbanked region (80h through 9Fh) cannot be accessed using banked addressing. The use of an unbanked SFR opcode is required to perform operations on these registers.

TABLE 3-1: ENC424J600/624J600 SFR MAP (SPI INTERFACE)

Banked Register Addresses	Bank 0 (00h offset)		Bank 1 (20h offset)		Bank 2 (40h offset)		Bank 3 (60h offset)		Unbanked[1] (80h offset)	
	Unbanked Address	Name	Unbanked Address	Name	Unbanked Address	Name	Unbanked Address	Name	Unbanked Address	Name
00	00	ETXSTL	20	EHT1L	40	MACON1L	60	MAADR3L	80	EGPDATA[2]
01	01	ETXSTH	21	EHT1H	41	MACON1H	61	MAADR3H	81	Reserved
02	02	ETXLENL	22	EHT2L	42	MACON2L	62	MAADR2L	82	ERXDATA[2]
03	03	ETXLENH	23	EHT2H	43	MACON2H	63	MAADR2H	83	Reserved
04	04	ERXSTL	24	EHT3L	44	MABBIPGL	64	MAADR1L	84	EUDADATA[2]
05	05	ERXSTH	25	EHT3H	45	MABBIPGH	65	MAADR1H	85	Reserved
06	06	ERXTAILL	26	EHT4L	46	MAIPGL	66	MIWRL	86	EGPRDPTL
07	07	ERXTAILH	27	EHT4H	47	MAIPGH	67	MIWRH	87	EGPRDPTH
08	08	ERXHEADL	28	EPMM1L	48	MACLCONL	68	MIRDL	88	EGPWRPTL
09	09	ERXHEADH	29	EPMM1H	49	MACLCONH	69	MIRDH	89	EGPWRPTH
0A	0A	EDMASTL	2A	EPMM2L	4A	MAMXFLL	6A	MISTATL	8A	ERXRDPTL
0B	0B	EDMASTH	2B	EPMM2H	4B	MAMXFLH	6B	MISTATH	8B	ERXRDPTH
0C	0C	EDMALENL	2C	EPMM3L	4C	Reserved	6C	EPAUSL	8C	ERXWRPTL
0D	0D	EDMALENH	2D	EPMM3H	4D	Reserved	6D	EPAUSH	8D	ERXWRPTH
0E	0E	EDMADSTL	2E	EPMM4L	4E	Reserved	6E	ECON2L	8E	EUDARDPTL
0F	0F	EDMADSTH	2F	EPMM4H	4F	Reserved	6F	ECON2H	8F	EUDARDPTH
10	10	EDMACSL	30	EPMCSL	50	Reserved	70	ERXWML	90	EUDAWRPTL
11	11	EDMACSH	31	EPMCSH	51	Reserved	71	ERXWMH	91	EUDAWRPTH
12	12	ETXSTATL	32	EPMOL	52	MICMDL	72	EIEL	92	Reserved
13	13	ETXSTATH	33	EPMOH	53	MICMDH	73	EIEH	93	Reserved
14	14	ETXWIREL	34	ERXFCONL	54	MIREGADRL	74	EIDLEDL	94	Reserved
15	15	ETXWIREH	35	ERXFCONH	55	MIREGADRH	75	EIDLEDH	95	Reserved
16	16	EUDASTL	36	EUDASTL	56	EUDASTL	76	EUDASTL	96	Reserved
17	17	EUDASTH	37	EUDASTH	57	EUDASTH	77	EUDASTH	97	Reserved
18	18	EUDANDL	38	EUDANDL	58	EUDANDL	78	EUDANDL	98	Reserved
19	19	EUDANDH	39	EUDANDH	59	EUDANDH	79	EUDANDH	99	Reserved
1A	1A	ESTATL	3A	ESTATL	5A	ESTATL	7A	ESTATL	9A	Reserved
1B	1B	ESTATH	3B	ESTATH	5B	ESTATH	7B	ESTATH	9B	Reserved
1C	1C	EIRL	3C	EIRL	5C	EIRL	7C	EIRL	9C	Reserved
1D	1D	EIRH	3D	EIRH	5D	EIRH	7D	EIRH	9D	Reserved
1E	1E	ECON1L	3E	ECON1L	5E	ECON1L	7E	ECON1L	9E	—
1F	1F	ECON1H	3F	ECON1H	5F	ECON1H	7F	ECON1H	9F	—

Note 1: Unbanked SFRs can be accessed only by unbanked SPI opcodes.

2: When using these registers to access the SRAM buffer, use only the N-byte SRAM instructions. See **Section 4.6.2 "Unbanked SFR Operations"** and **Section 4.6.3 "SRAM Buffer Operations"** for more details.

3.2.4 PSP REGISTER MAP

When using a PSP interface, the SFR memory is linear; all registers are directly accessible without banking. To maintain consistency with the SPI interface, the EUDAST, EUDAND, ESTAT, EIR and ECON1 registers are instantiated in four locations in the PSP memory maps. Users may opt to use any one of these four locations.

The SFR maps for the 8-bit and 16-bit PSP interfaces are shown in Table 3-2 and Table 3-3, respectively.

TABLE 3-2: ENC424J600/624J600 SFR MAP (BASE REGISTER MAP, 8-BIT PSP INTERFACE)

Addr	Name	Addr	Name	Addr	Name	Addr	Name	Addr	Name
7E00	ETXSTL	7E20	EHT1L	7E40	MACON1L	7E60	MAADR3L	7E80	EGPDATA
7E01	ETXSTH	7E21	EHT1H	7E41	MACON1H	7E61	MAADR3H	7E81	Reserved
7E02	ETXLENL	7E22	EHT2L	7E42	MACON2L	7E62	MAADR2L	7E82	ERXDATA
7E03	ETXLENH	7E23	EHT2H	7E43	MACON2H	7E63	MAADR2H	7E83	Reserved
7E04	ERXSTL	7E24	EHT3L	7E44	MABBIPGL	7E64	MAADR1L	7E84	EUDADATA
7E05	ERXSTH	7E25	EHT3H	7E45	MABBIPGH	7E65	MAADR1H	7E85	Reserved
7E06	ERXTAILL	7E26	EHT4L	7E46	MAIPGL	7E66	MIWRL	7E86	EGPRDPTL
7E07	ERXTAILH	7E27	EHT4H	7E47	MAIPGH	7E67	MIWRH	7E87	EGPRDPTH
7E08	ERXHEADL	7E28	EPMM1L	7E48	MACLCONL	7E68	MIRDL	7E88	EGPWRPTL
7E09	ERXHEADH	7E29	EPMM1H	7E49	MACLCONH	7E69	MIRDH	7E89	EGPWRPTH
7E0A	EDMASTL	7E2A	EPMM2L	7E4A	MAMXFLL	7E6A	MISTATL	7E8A	ERXRDPTL
7E0B	EDMASTH	7E2B	EPMM2H	7E4B	MAMXFLH	7E6B	MISTATH	7E8B	ERXRDPTH
7E0C	EDMALENL	7E2C	EPMM3L	7E4C	Reserved	7E6C	EPAUSL	7E8C	ERXWRPTL
7E0D	EDMALENH	7E2D	EPMM3H	7E4D	Reserved	7E6D	EPAUSH	7E8D	ERXWRPTH
7E0E	EDMADSTL	7E2E	EPMM4L	7E4E	Reserved	7E6E	ECON2L	7E8E	EUDARDPTL
7E0F	EDMADSTH	7E2F	EPMM4H	7E4F	Reserved	7E6F	ECON2H	7E8F	EUDARDPTH
7E10	EDMACSL	7E30	EPMCSL	7E50	Reserved	7E70	ERXWML	7E90	EUDAWRPTL
7E11	EDMACSH	7E31	EPMCSH	7E51	Reserved	7E71	ERXWMH	7E91	EUDAWRPTH
7E12	ETXSTATL	7E32	EPMOL	7E52	MICMDL	7E72	EIEL	7E92	Reserved
7E13	ETXSTATH	7E33	EPMOH	7E53	MICMDH	7E73	EIEH	7E93	Reserved
7E14	ETXWIREL	7E34	ERXFCONL	7E54	MIREGADRL	7E74	EIDLEDL	7E94	Reserved
7E15	ETXWIREH	7E35	ERXFCONH	7E55	MIREGADRH	7E75	EIDLEDH	7E95	Reserved
7E16	EUDASTL	7E36	EUDASTL	7E56	EUDASTL	7E76	EUDASTL	7E96	Reserved
7E17	EUDASTH	7E37	EUDASTH	7E57	EUDASTH	7E77	EUDASTH	7E97	Reserved
7E18	EUDANDL	7E38	EUDANDL	7E58	EUDANDL	7E78	EUDANDL	7E98	Reserved
7E19	EUDANDH	7E39	EUDANDH	7E59	EUDANDH	7E79	EUDANDH	7E99	Reserved
7E1A	ESTATL	7E3A	ESTATL	7E5A	ESTATL	7E7A	ESTATL	7E9A	Reserved
7E1B	ESTATH	7E3B	ESTATH	7E5B	ESTATH	7E7B	ESTATH	7E9B	Reserved
7E1C	EIRL	7E3C	EIRL	7E5C	EIRL	7E7C	EIRL	7E9C	Reserved
7E1D	EIRH	7E3D	EIRH	7E5D	EIRH	7E7D	EIRH	7E9D	Reserved
7E1E	ECON1L	7E3E	ECON1L	7E5E	ECON1L	7E7E	ECON1L	7E9E	—
7E1F	ECON1H	7E3F	ECON1H	7E5F	ECON1H	7E7F	ECON1H	7E9F	—

TABLE 3-3: ENC424J600/624J600 SFR MAP (BASE REGISTER MAP, 16-BIT PSP INTERFACE)

Addr	Name	Addr	Name	Addr	Name	Addr	Name	Addr	Name
3F00	ETXST	3F10	EHT1	3F20	MACON1	3F30	MAADR3	3F40	EGPDATA
3F01	ETXLEN	3F11	EHT2	3F21	MACON2	3F31	MAADR2	3F41	ERXDATA
3F02	ERXST	3F12	EHT3	3F22	MABBIPG	3F32	MAADR1	3F42	EUDADATA
3F03	ERXTAIL	3F13	EHT4	3F23	MAIPG	3F33	MIWR	3F43	EGPRDPT
3F04	ERXHEAD	3F14	EPMM1	3F24	MACLCON	3F34	MIRD	3F44	EGPWRPT
3F05	EDMAST	3F15	EPMM2	3F25	MAMXFL	3F35	MISTAT	3F45	ERXRDPT
3F06	EDMALEN	3F16	EPMM3	3F26	Reserved	3F36	EPAUS	3F46	ERXWRPT
3F07	EDMADST	3F17	EPMM4	3F27	Reserved	3F37	ECON2	3F47	EUDARDPT
3F08	EDMACS	3F18	EPMCS	3F28	Reserved	3F38	ERXWM	3F48	EUDAWRPT
3F09	ETXSTAT	3F19	EPMO	3F29	MICMD	3F39	EIE	3F49	Reserved
3F0A	ETXWIRE	3F1A	ERXFCON	3F2A	MIREGADR	3F3A	EIDLED	3F4A	Reserved
3F0B	EUDAST	3F1B	EUDAST	3F2B	EUDAST	3F3B	EUDAST	3F4B	Reserved
3F0C	EUDAND	3F1C	EUDAND	3F2C	EUDAND	3F3C	EUDAND	3F4C	Reserved
3F0D	ESTAT	3F1D	ESTAT	3F2D	ESTAT	3F3D	ESTAT	3F4D	Reserved
3F0E	EIR	3F1E	EIR	3F2E	EIR	3F3E	EIR	3F4E	Reserved
3F0F	ECON1	3F1F	ECON1	3F2F	ECON1	3F3F	ECON1	3F4F	—

3.2.4.1 PSP Bit Set and Bit Clear Registers

A major difference between the SPI and PSP memory maps is the inclusion of companion Bit Set and Bit Clear registers for many of the E registers. Since the PSP interface allows direct access to memory locations, without a command interpreter, there are no instructions implemented to perform single bit manipulations. Instead, this interface implements separate Bit Set and Bit Clear registers, allowing users to individually work with volatile bits (such as interrupt flags) without the risk of disturbing the values of other bits. Setting the bit(s) in one of these registers sets or clears the corresponding bit(s) in the base register.

In the PSP interface, Bit Set and Bit Clear registers are located in different areas of the addressable memory space from their corresponding “base” SFRs. The address of the registers is always at a fixed offset from their corresponding base register. For the 8-bit interface, the offset is 100h (Set) or 180h (Clear). For the 16-bit interface, the offset is 80H (Set) or C0 (Clear). Symbolically, the names of the companion registers are the names of the base registers, plus the suffix form “-SET” (or “-SETH/SETL”) for Bit Set registers and “-CLR” (“-CLRH/CLRL”) for Bit Clear registers.

Most SFRs have their own pair of Bit Set and Bit Clear registers. However, these SFRs do not:

- MAC registers, including MI registers for PHY access
- Read-only status registers (ERXHEAD, ETXSTAT, ETXWIRE and ESTAT)
- All of the SRAM Buffer Pointers and data windows (SFRs located at 7E80h to 7E9Fh in the 8-bit interface, or 3F40h to 3F4Fh in the 16-bit interface)

The Bit Set and Bit Clear registers for the 8-bit PSP interface are listed in Table 3-4 and Table 3-5, respectively. The registers for the 16-bit interface are listed together in Table 3-6.

TABLE 3-4: ENC424J600/624J600 SFR MAP (SET REGISTER MAP, 8-BIT PSP INTERFACE)

Bit Set Registers (7F00h to 7F7Fh)[1]							
Addr	**Name**	**Addr**	**Name**	**Addr**	**Name**	**Addr**	**Name**
7F00	ETXSTSETL	7F20	EHT1SETL	7F40	Reserved	7F60	Reserved
7F01	ETXSTSETH	7F21	EHT1SETH	7F41	Reserved	7F61	Reserved
7F02	ETXLENSETL	7F22	EHT2SETL	7F42	Reserved	7F62	Reserved
7F03	ETXLENSETH	7F23	EHT2SETH	7F43	Reserved	7F63	Reserved
7F04	ERXSTSETL	7F24	EHT3SETL	7F44	Reserved	7F64	Reserved
7F05	ERXSTSETH	7F25	EHT3SETH	7F45	Reserved	7F65	Reserved
7F06	ERXTAILSETL	7F26	EHT4SETL	7F46	Reserved	7F66	Reserved
7F07	ERXTAILSETH	7F27	EHT4SETH	7F47	Reserved	7F67	Reserved
7F08	—	7F28	EPMM1SETL	7F48	Reserved	7F68	Reserved
7F09	—	7F29	EPMM1SETH	7F49	Reserved	7F69	Reserved
7F0A	EDMASTSETL	7F2A	EPMM2SETL	7F4A	Reserved	7F6A	Reserved
7F0B	EDMASTSETH	7F2B	EPMM2SETH	7F4B	Reserved	7F6B	Reserved
7F0C	EDMALENSETL	7F2C	EPMM3SETL	7F4C	Reserved	7F6C	EPAUSSETL
7F0D	EDMALENSETH	7F2D	EPMM3SETH	7F4D	Reserved	7F6D	EPAUSSETH
7F0E	EDMADSTSETL	7F2E	EPMM4SETL	7F4E	Reserved	7F6E	ECON2SETL
7F0F	EDMADSTSETH	7F2F	EPMM4SETH	7F4F	Reserved	7F6F	ECON2SETH
7F10	EDMACSSETL	7F30	EPMCSSETL	7F50	Reserved	7F70	ERXWMSETL
7F11	EDMACSSETH	7F31	EPMCSSETH	7F51	Reserved	7F71	ERXWMSETH
7F12	—	7F32	EPMOSETL	7F52	Reserved	7F72	EIESETL
7F13	—	7F33	EPMOSETH	7F53	Reserved	7F73	EIESETH
7F14	—	7F34	ERXFCONSETL	7F54	Reserved	7F74	EIDLEDSETL
7F15	—	7F35	ERXFCONSETH	7F55	Reserved	7F75	EIDLEDSETH
7F16	EUDASTSETL	7F36	EUDASTSETL	7F56	EUDASTSETL	7F76	EUDASTSETL
7F17	EUDASTSETH	7F37	EUDASTSETH	7F57	EUDASTSETH	7F77	EUDASTSETH
7F18	EUDANDSETL	7F38	EUDANDSETL	7F58	EUDANDSETL	7F78	EUDANDSETL
7F19	EUDANDSETH	7F39	EUDANDSETH	7F59	EUDANDSETH	7F79	EUDANDSETH
7F1A	—	7F3A	—	7F5A	—	7F7A	—
7F1B	—	7F3B	—	7F5B	—	7F7B	—
7F1C	EIRSETL	7F3C	EIRSETL	7F5C	EIRSETL	7F7C	EIRSETL
7F1D	EIRSETH	7F3D	EIRSETH	7F5D	EIRSETH	7F7D	EIRSETH
7F1E	ECON1SETL	7F3E	ECON1SETL	7F5E	ECON1SETL	7F7E	ECON1SETL
7F1F	ECON1SETH	7F3F	ECON1SETH	7F5F	ECON1SETH	7F7F	ECON1SETH

Note 1: Bit Set and Bit Clear registers are not implemented for the base SFRs located between 7E80h and 7E9Fh.

TABLE 3-5: ENC424J600/624J600 SFR MAP (CLR REGISTER MAP, 8-BIT PSP INTERFACE)

Bit Clear Registers (7F80h to 7FFFh)(1)							
Addr	**Name**	**Addr**	**Name**	**Addr**	**Name**	**Addr**	**Name**
7F80	ETXSTCLRL	7FA0	EHT1CLRL	7FC0	Reserved	7FE0	Reserved
7F81	ETXSTCLRH	7FA1	EHT1CLRH	7FC1	Reserved	7FE1	Reserved
7F82	ETXLENCLRL	7FA2	EHT2CLRL	7FC2	Reserved	7FE2	Reserved
7F83	ETXLENCLRH	7FA3	EHT2CLRH	7FC3	Reserved	7FE3	Reserved
7F84	ERXSTCLRL	7FA4	EHT3CLRL	7FC4	Reserved	7FE4	Reserved
7F85	ERXSTCLRH	7FA5	EHT3CLRH	7FC5	Reserved	7FE5	Reserved
7F86	ERXTAILCLRL	7FA6	EHT4CLRL	7FC6	Reserved	7FE6	Reserved
7F87	ERXTAILCLRH	7FA7	EHT4CLRH	7FC7	Reserved	7FE7	Reserved
7F88	—	7FA8	EPMM1CLRL	7FC8	Reserved	7FE8	Reserved
7F89	—	7FA9	EPMM1CLRH	7FC9	Reserved	7FE9	Reserved
7F8A	EDMASTCLRL	7FAA	EPMM2CLRL	7FCA	Reserved	7FEA	Reserved
7F8B	EDMASTCLRH	7FAB	EPMM2CLRH	7FCB	Reserved	7FEB	Reserved
7F8C	EDMALENCLRL	7FAC	EPMM3CLRL	7FCC	Reserved	7FEC	EPAUSCLRL
7F8D	EDMALENCLRH	7FAD	EPMM3CLRH	7FCD	Reserved	7FED	EPAUSCLRH
7F8E	EDMADSTCLRL	7FAE	EPMM4CLRL	7FCE	Reserved	7FEE	ECON2CLRL
7F8F	EDMADSTCLRH	7FAF	EPMM4CLRH	7FCF	Reserved	7FEF	ECON2CLRH
7F90	EDMACSCLRL	7FB0	EPMCSCLRL	7FD0	Reserved	7FF0	ERXWMCLRL
7F91	EDMACSCLRH	7FB1	EPMCSCLRH	7FD1	Reserved	7FF1	ERXWMCLRH
7F92	—	7FB2	EPMOCLRL	7FD2	Reserved	7FF2	EIECLRL
7F93	—	7FB3	EPMOCLRH	7FD3	Reserved	7FF3	EIECLRH
7F94	—	7FB4	ERXFCONCLRL	7FD4	Reserved	7FF4	EIDLEDCLRL
7F95	—	7FB5	ERXFCONCLRH	7FD5	Reserved	7FF5	EIDLEDCLRH
7F96	EUDASTCLRL	7FB6	EUDASTCLRL	7FD6	EUDASTCLRL	7FF6	EUDASTCLRL
7F97	EUDASTCLRH	7FB7	EUDASTCLRH	7FD7	EUDASTCLRH	7FF7	EUDASTCLRH
7F98	EUDANDCLRL	7FB8	EUDANDCLRL	7FD8	EUDANDCLRL	7FF8	EUDANDCLRL
7F99	EUDANDCLRH	7FB9	EUDANDCLRH	7FD9	EUDANDCLRH	7FF9	EUDANDCLRH
7F9A	—	7FBA	—	7FDA	—	7FFA	—
7F9B	—	7FBB	—	7FDB	—	7FFB	—
7F9C	EIRCLRL	7FBC	EIRCLRL	7FDC	EIRCLRL	7FFC	EIRCLRL
7F9D	EIRCLRH	7FBD	EIRCLRH	7FDD	EIRCLRH	7FFD	EIRCLRH
7F9E	ECON1CLRL	7FBE	ECON1CLRL	7FDE	ECON1CLRL	7FFE	ECON1CLRL
7F9F	ECON1CLRH	7FBF	ECON1CLRH	7FDF	ECON1CLRH	7FFF	ECON1CLRH

Note 1: Bit Set and Bit Clear registers are not implemented for the base SFRs located between 7E80h and 7E9Fh.

TABLE 3-6: ENC424J600/624J600 SFR MAP (SET/CLR REGISTER MAP, 16-BIT PSP INTERFACE)

Bit Set Registers (3F80h to 3FBFh)(1)							
Addr	**Name**	**Addr**	**Name**	**Addr**	**Name**	**Addr**	**Name**
3F80	ETXSTSET	3F90	EHT1SET	3FA0	Reserved	3FB0	Reserved
3F81	ETXLENSET	3F91	EHT2SET	3FA1	Reserved	3FB1	Reserved
3F82	ERXSTSET	3F92	EHT3SET	3FA2	Reserved	3FB2	Reserved
3F83	ERXTAILSET	3F93	EHT4SET	3FA3	Reserved	3FB3	Reserved
3F84	—	3F94	EPMM1SET	3FA4	Reserved	3FB4	Reserved
3F85	EDMASTSET	3F95	EPMM2SET	3FA5	Reserved	3FB5	Reserved
3F86	EDMALENSET	3F96	EPMM3SET	3FA6	Reserved	3FB6	EPAUSSET
3F87	EDMADSTSET	3F97	EPMM4SET	3FA7	Reserved	3FB7	ECON2SET
3F88	EDMACSSET	3F98	EPMCSSET	3FA8	Reserved	3FB8	ERXWMSET
3F89	—	3F99	EPMOSET	3FA9	Reserved	3FB9	EIESET
3F8A	—	3F9A	ERXFCON	3FAA	Reserved	3FBA	EIDLEDSET
3F8B	EUDASTSET	3F9B	EUDASTSET	3FAB	EUDASTSET	3FBB	EUDASTSET
3F8C	EUDANDSET	3F9C	EUDANDSET	3FAC	EUDANDSET	3FBC	EUDANDSET
3F8D	—	3F9D	—	3FAD	—	3FBD	—
3F8E	EIRSET	3F9E	EIRSET	3FAE	EIRSET	3FBE	EIRSET
3F8F	ECON1SET	3F9F	ECON1SET	3FAF	ECON1SET	3FBF	ECON1SET
Bit Clear Registers (3FC0h to 3FFFh)(1)							
Addr	**Name**	**Addr**	**Name**	**Addr**	**Name**	**Addr**	**Name**
3FC0	ETXSTCLR	3FD0	EHT1CLR	3FE0	Reserved	3FF0	Reserved
3FC1	ETXLENCLR	3FD1	EHT2CLR	3FE1	Reserved	3FF1	Reserved
3FC2	ERXSTCLR	3FD2	EHT3CLR	3FE2	Reserved	3FF2	Reserved
3FC3	ERXTAILCLR	3FD3	EHT4CLR	3FE3	Reserved	3FF3	Reserved
3FC4	—	3FD4	EPMM1CLR	3FE4	Reserved	3FF4	Reserved
3FC5	EDMASTCLR	3FD5	EPMM2CLR	3FE5	Reserved	3FF5	Reserved
3FC6	EDMALENCLR	3FD6	EPMM3CLR	3FE6	Reserved	3FF6	EPAUSCLR
3FC7	EDMADSTCLR	3FD7	EPMM4CLR	3FE7	Reserved	3FF7	ECON2CLR
3FC8	EDMACSCLR	3FD8	EPMCSCLR	3FE8	Reserved	3FF8	ERXWMCLR
3FC9	—	3FD9	EPMOCLR	3FE9	Reserved	3FF9	EIECLR
3FCA	—	3FDA	ERXFCONCLR	3FEA	Reserved	3FFA	EIDLEDCLR
3FCB	EUDASTCLR	3FDB	EUDASTCLR	3FEB	EUDASTCLR	3FFB	EUDASTCLR
3FCC	EUDANDCLR	3FDC	EUDANDCLR	3FEC	EUDANDCLR	3FFC	EUDANDCLR
3FCD	—	3FDD	—	3FED	—	3FFD	—
3FCE	EIRCLR	3FDE	EIRCLR	3FEE	EIRCLR	3FFE	EIRCLR
3FCF	ECON1CLR	3FDF	ECON1CLR	3FEF	ECON1CLR	3FFF	ECON1CLR

Note 1: Bit Set and Bit Clear registers are not implemented for the base SFRs located between 3F40h and 3F4Fh.

TABLE 3-7: ENC424J600/624J600 REGISTER FILE SUMMARY

File Name	8-Bit	High Byte ('H' Register)								Low Byte ('L' Register)								Reset
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
	16-Bit	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
EUDAST		—	User-Defined Area Start Pointer (EUDAST<14:8>)							User-Defined Area Start Pointer (EUDAST<7:0>)								00, 00
EUDAND		—	User-Defined Area End Pointer (EUDAND<14:8>)							User-Defined Area End Pointer (EUDAND<7:0>)								5F, FF
ESTAT		INT	FCIDLE	RXBUSY	CLKRDY	r	PHYDPX	r	PHYLNK	PKTCNT7	PKTCNT6	PKTCNT5	PKTCNT4	PKTCNT3	PKTCNT2	PKTCNT1	PKTCNT0	00, 00
EIR		CRYPTEN	MODEXIF	HASHIF	AESIF	LINKIF	r	r	r	r	PKTIF	DMAIF	r	TXIF	TXABTIF	RXABTIF	PCFULIF	0A, 00
ECON1		MODEXST	HASHEN	HASHOP	HASHLST	AESST	AESOP1	AESOP0	PKTDEC	FCOP1	FCOP0	DMAST	DMACPY	DMACSSD	DMANOCS	TXRTS	RXEN	00, 00
ETXST		—	TX Start Address (ETXST<14:8>)							TX Start Address (ETXST<7:0>)								00, 00
ETXLEN		—	TX Length (ETXLEN<14:8>)							TX Length (ETXLEN<7:0>)								00, 00
ERXST		—	RX Buffer Start Address (ERXST<14:8>)							RX Buffer Start Address (ERXST<7:0>)								53, 40
ERXTAIL		—	RX Tail Pointer (ERXTAIL<14:8>)							RX Tail Pointer (ERXTAIL<7:0>)								5F, FE
ERXHEAD		—	RX Head Pointer (ERXHEAD<14:8>)							RX Head Pointer (ERXHEAD<7:0>)								53, 40
EDMAST		—	DMA Start Address (EDMAST<14:8>)							DMA Start Address (EDMAST<7:0>)								00, 00
EDMALEN		—	DMA Length (EDMALEN<14:8>)							DMA Length (EDMALEN<7:0>)								00, 00
EDMADST		—	DMA Destination Address (EDMADST<14:8>)							DMA Destination Address (EDMADST<7:0>)								00, 00
EDMACS		DMA Checksum, High Byte (EDMACS<15:8>)								DMA Checksum, Low Byte (EDMACS<7:0>)								00, 00
ETXSTAT		—	—	—	r	r	LATECOL	MAXCOL	EXDEFER	DEFER	r	r	CRCBAD	COLCNT3	COLCNT2	COLCNT1	COLCNT0	00, 00
ETXWIRE		Transmit Byte Count on Wire (including collision bytes), High Byte (ETXWIRE<15:8>)								Transmit Byte Count on Wire (including collision bytes), Low Byte (ETXWIRE<7:0>)								00, 00
EHT1		Hash Table Filter (EHT1<15:8>)								Hash Table Filter (EHT1<7:0>)								00, 00
EHT2		Hash Table Filter (EHT2<31:24>)								Hash Table Filter (EHT2<23:16>)								00, 00
ETH3		Hash Table Filter (EHT3<47:40>)								Hash Table Filter (EHT3<39:32>)								00, 00
ETH4		Hash Table Filter (EHT4<63:56>)								Hash Table Filter (EHT4<55:48>)								00, 00
EPMM1		Pattern Match Filter Mask (EPMM1<15:8>)								Pattern Match Filter Mask (EPMM1<7:0>)								00, 00
EPMM2		Pattern Match Filter Mask (EPMM2<15:8>)								Pattern Match Filter Mask (EPMM2<7:0>)								00, 00
EPMM3		Pattern Match Filter Mask (EPMM3<15:8>)								Pattern Match Filter Mask (EPMM3<7:0>)								00, 00
EPMM4		Pattern Match Filter Mask (EPMM4<15:8>)								Pattern Match Filter Mask (EPMM4<7:0>)								00, 00
EPMCS		Pattern Match Filter Checksum, High Byte (EPMCS<15:8>)								Pattern Match Filter Checksum, Low Byte (EPMCS<7:0>)								00, 00
ERXFCON		HTEN	MPEN	—	NOTPM	PMEN3	PMEN2	PMEN1	PMEN0	CRCEEN	CRCEN	RUNTEEN	RUNTEN	UCEN	NOTMEEN	MCEN	BCEN	00, 59
EPMO		Pattern Match Filter Offset, High Byte (EPMO<15:8>)								Pattern Match Filter Offset, Low Byte (EPMO<7:0>)								00, 00
MACON1		r	r	—	—	r	r	r	r	—	—	—	LOOPBK	r	RXPAUS	PASSALL	r	x0, 0D
MACON2		—	DEFER	BPEN	NOBKOFF	—	—	r	r	PADCFG2	PADCFG1	PADCFG0	TXCRCEN	PHDREN	HFRMEN	r	FULDPX	40, B2
MABBIPG		—	—	—	—	—	—	—	—	—	BBIPG6	BBIPG5	BBIPG4	BBIPG3	BBIPG2	BBIPG1	BBIPG0	00, 12
MAIPG		—	r	r	r	r	r	r	r	—	IPG6	IPG5	IPG4	IPG3	IPG2	IPG1	IPG0	0C, 12
MACLCON		—	—	r	r	r	r	r	r	—	—	—	—	MAXRET3	MAXRET2	MAXRET1	MAXRET0	37, 0F
MAMXFL		MAC Maximum Frame Length, High Byte (MAMXFL<15:8>)								MAC Maximum Frame Length, Low Byte (MAMXFL<7:0>)								05, EE

Legend: — = unimplemented, read as '0'; q = unique MAC address or silicon revision nibble; r = reserved bit, do not modify; x = Reset value unknown. Reset values are shown in hexadecimal for each byte.

TABLE 3-7: ENC424J600/624J600 REGISTER FILE SUMMARY (CONTINUED)

File Name	8-Bit	High Byte ('H' Register)								Low Byte ('L' Register)								Reset
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
	16-Bit	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MICMD		—	—	—	—	—	—	—	—	—	—	—	—	—	—	MIISCAN	MIIRD	--, 00
MIREGADR		—	—	—	r	r	r	r	r	—	—	—	PHREG4	PHREG3	PHREG2	PHREG1	PHREG0	01, 00
MAADR3		MAC Address, Byte 6 (MAADR<7:0>)								MAC Address, Byte 5 (MAADR<15:8>)								qq, qq
MAADR2		MAC Address, Byte 4 (MAADR<23:16>)								MAC Address, Byte 3 (MAADR<31:24>)/OUI Byte 3								qq, a3
MAADR1		MAC Address, Byte 2 (MAADR<39:32>)/OUI Byte 2								MAC Address, Byte 1 (MAADR<47:40>)/OUI Byte 1								04, 00
MIWR		MII Management Write Data, High Byte (MIWR<15:8>)								MII Management Write Data, Low Byte (MIWR<7:0>)								00, 00
MIRD		MII Management Read Data, High Byte (MIRD<15:8>)								MII Management Read Data, Low Byte (MIRD<7:0>)								00, 00
MISTAT		—	—	—	—	—	—	—	—	—	—	—	—	r	NVALID	SCAN	BUSY	--, 00
EPAUS		Pause Timer Value, High Byte (EPAUS<15:8>)								Pause Timer Value, Low Byte (EPAUS<7:0>)								10, 00
ECON2		ETHEN	STRCH	TXMAC	SHA1MD5	COCON3	COCON2	COCON1	COCON0	AUTOFC	TXRST	RXRST	ETHRST	MODLEN1	MODLEN0	AESLEN1	AESLEN0	CB, 00
ERXWM		RXFWM7	RXFWM6	RXFWM5	RXFWM4	RXFWM3	RXFWM2	RXFWM1	RXFWM0	RXEWM7	RXEWM6	RXEWM5	RXEWM4	RXEWM3	RXEWM2	RXEWM1	RXEWM0	10, 0F
EIE		INTIE	MODEXIE	HASHIE	AESIE	LINKIE	r	r	r	r	PKTIE	DMAIE	r	TXIE	TXABTIE	RXABTIE	PCFULIE	80, 10
EIDLED		LACFG3	LACFG2	LACFG1	LACFG0	LBCFG3	LBCFG2	LBCFG1	LBCFG0	DEVID2	DEVID1	DEVID0	REVID4	REVID3	REVID2	REVID1	REVID0	26, qq
EGPDATA		r	r	r	r	r	r	r	r	General Purpose Data Window Register								--, xx
ERXDATA		r	r	r	r	r	r	r	r	Ethernet RX Data Window Register								--, xx
EUDADATA		r	r	r	r	r	r	r	r	User-Defined Area Data Window Register								--, xx
EGPRDPT		—	General Purpose Window Read Pointer, High Byte (ETXRDPT<14:8>)							General Purpose Window Read Pointer, Low Byte (ETXRDPT<7:0>)								05, FA
EGPWRPT		—	General Purpose Window Write Pointer, High Byte (ETXWRPT<14:8>)							General Purpose Window Write Pointer, Low Byte (ETXWRPT<7:0>)								00, 00
ERXRDPT		—	RX Window Read Pointer, High Byte (ERXRDPT<14:8>)							RX Window Read Pointer, Low Byte (ERXRDPT<7:0>)								05, FA
ERXWRPT		—	RX Window Write Pointer, High Byte (ERXWRPT<14:8>)							RX Window Write Pointer, Low Byte (ERXWRPT<7:0>)								00, 00
EUDARDPT		—	UDA Window Read Pointer (EUDARDPT<14:8>)							UDA Window Read Pointer (EUDARDPT<7:0>)								05, FA
EUDAWRPT		—	UDA Window Write Pointer (EUDAWRPT<14:8>)							UDA Window Write Pointer (EUDAWRPT<7:0>)								00, 00

Legend: — = unimplemented, read as '0'; q = unique MAC address or silicon revision nibble; r = reserved bit, do not modify; x = Reset value unknown. Reset values are shown in hexadecimal for each byte.

3.3 PHY Special Function Registers

The PHY registers provide configuration and control of the PHY module, as well as status information about its operation. These 16-bit registers are located in their own memory space, outside of the main SFR space.

Unlike other SFRs, the PHY SFRs are not directly accessible through the SPI or PSP interfaces. Instead, access is accomplished through a special set of MAC control registers that implement a Media Independent Interface Management (MIIM) defined by IEEE 802.3; these are the MICMD, MISTAT and MIREGADR registers.

There are a total of 32 PHY addresses; however, only 10 locations implement user-accessible registers listed in Table 3-8. Writes to unimplemented locations are ignored and any attempts to read these locations return FFFFh. Do not write to reserved PHY register locations and ignore their content if read.

TABLE 3-8: PHY SPECIAL FUNCTION REGISTER MAP

Address	Name	Address	Name
00	PHCON1	10	Reserved
01	PHSTAT1	11	PHCON2
02	Reserved	12	Reserved
03	Reserved	13	—
04	PHANA	14	Reserved
05	PHANLPA	15	Reserved
06	PHANE	16	Reserved
07	—	17	Reserved
08	—	18	—
09	—	19	—
0A	—	1A	—
0B	—	1B	PHSTAT2
0C	—	1C	Reserved
0D	—	1D	Reserved
0E	—	1E	Reserved
0F	—	1F	PHSTAT3

3.3.1 READING PHY REGISTERS

When a PHY register is read, the entire 16 bits are obtained.

To read from a PHY register:

1. Write the address of the PHY register to read from into the MIREGADR register (Register 3-1). Make sure to also set reserved bit 8 of this register.
2. Set the MIIRD bit (MICMD<0>, Register 3-2). The read operation begins and the BUSY bit (MISTAT<0>, Register 3-3) is automatically set by hardware.
3. Wait 25.6 μs. Poll the BUSY (MISTAT<0>) bit to be certain that the operation is complete. While busy, the host controller should not start any MIISCAN operations or write to the MIWR register. When the MAC has obtained the register contents, the BUSY bit will clear itself.
4. Clear the MIIRD (MICMD<0>) bit.
5. Read the desired data from the MIRD register. For 8-bit interfaces, the order that these bytes are read is unimportant.

3.3.2 WRITING PHY REGISTERS

When a PHY register is written to, the entire 16 bits are written at once; selective bit writes are not implemented. If it is necessary to reprogram only select bits in the register, the host microcontroller must first read the PHY register, modify the resulting data and then write the data back to the PHY register.

To write to a PHY register:

1. Write the address of the PHY register to write to into the MIREGADR register. Make sure to also set reserved bit 8 of this register.
2. Write the 16 bits of data into the MIWR register. The low byte must be written first, followed by the high byte.
3. Writing to the high byte of MIWR begins the MIIM transaction and the BUSY (MISTAT<0>) bit is automatically set by hardware.

The PHY register is written after the MIIM operation completes, which takes 25.6 μs. When the write operation has completed, the BUSY bit clears itself. The host controller should not start any MIISCAN, MIWR or MIIRD operations while the BUSY bit is set.

3.3.3 SCANNING A PHY REGISTER

The MAC can be configured to perform automatic back-to-back read operations on a PHY register. This can reduce the host controller complexity when periodic status information updates are desired.

To perform the scan operation:

1. Write the address of the PHY register to read from into the MIREGADR register. Make sure to also set reserved bit 8 of this register.
2. Set the MIISCAN (MICMD<1>) bit. The scan operation begins and the BUSY (MISTAT<0>) bit is automatically set by hardware. The first read operation will complete after 25.6 μs. Subsequent reads will be done at the same interval until the operation is cancelled. The NVALID (MISTAT<2>) bit may be polled to determine when the first read operation is complete.

After setting the MIISCAN bit, the MIRD register will automatically be updated every 25.6 μs. There is no status information which can be used to determine when the MIRD registers are updated. On the SPI or 8-bit PSP interfaces, the host controller can only read one register location at a time. Therefore, it must not be assumed that the values of MIRDL and MIRDH were read from the PHY at exactly the same time.

When the MIISCAN operation is in progress, the host controller must not attempt to write to MIWR or start an MIIRD operation. The MIISCAN operation can be cancelled by clearing the MIISCAN (MICMD<1>) bit and then polling the BUSY (MISTAT<0>) bit. New operations may be started after the BUSY bit is cleared.

REGISTER 3-1: MIREGADR: MII MANAGEMENT ADDRESS REGISTER

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
—	—	—	r	r	r	r	r
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	PHREG4	PHREG3	PHREG2	PHREG1	PHREG0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **Reserved:** Write as ‘00001’ (01h)

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **PHREG<4:0>:** MII Management PHY Register Address Select bits
The address of the PHY register which MII Management read and write operations will apply to.

REGISTER 3-2: MICMD: MII MANAGEMENT COMMAND REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	MIISCAN	MIIRD
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-2 **Unimplemented:** Read as '0'

bit 1 **MIISCAN:** MII Scan Enable bit
1 = PHY register designated by MIREGADR<4:0> is continuously read and the data is copied to MIRD
0 = No MII Management scan operation is in progress

bit 0 **MIIRD:** MII Read Enable bit
1 = PHY register designated by MIREGADR<4:0> is read once and the data is copied to MIRD
0 = No MII Management read operation is in progress

REGISTER 3-3: MISTAT: MII MANAGEMENT STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—	r	NVALID	SCAN	BUSY
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-4 **Unimplemented:** Read as '0'

bit 3 **Reserved:** Ignore on read

bit 2 **NVALID:** MII Management Read Data Not Valid Status bit
1 = The contents of MIRD are not valid yet
0 = The MII Management read cycle has completed and MIRD has been updated

bit 1 **SCAN:** MII Management Scan Status bit
1 = MII Management scan operation is in progress
0 = No MII Management scan operation is in progress

bit 0 **BUSY:** MII Management Busy Status bit
1 = A PHY register is currently being read or written to
0 = The MII Management interface is Idle

TABLE 3-9: PHY REGISTER FILE SUMMARY

File Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Reset
PHCON1	PRST	PLOOPBK	SPD100	ANEN	PSLEEP	r	RENEG	PFULDPX	r	r	r	r	r	r	r	r	10, 00
PHSTAT1	r	FULL100	HALF100	FULL10	HALF10	r	r	r	r	r	ANDONE	LRFAULT	ANABLE	LLSTAT	r	EXTREGS	78, 09
PHANA	ADNP	r	ADFAULT	r	ADPAUS1	ADPAUS0	r	AD100FD	AD100	AD10FD	AD10	ADIEEE4	ADIEEE3	ADIEEE2	ADIEEE1	ADIEEE0	01, E1
PHANLPA	LPNP	LPACK	LPFAULT	r	LPPAUS1	LPPAUS0	LP100T4	LP100FD	LP100	LP10FD	LP10	LPIEEE4	LPIEEE3	LPIEEE2	LPIEEE1	LPIEEE0	xx, xx
PHANE	r	r	r	r	r	r	r	r	r	r	r	PDFLT	r	r	LPARCD	LPANABL	00, 00
PHCON2	r	r	EDPWRDN	r	EDTHRES	r	r	r	r	r	r	r	r	FRCLNK	EDSTAT	r	00, 02
PHSTAT2	r	r	r	r	r	r	r	r	r	r	r	PLRITY	r	r	r	r	xx, 0x
PHSTAT3	r	r	r	r	r	r	r	r	r	r	r	SPDDPX2	SPDDPX1	SPDDPX0	r	r	00, 40

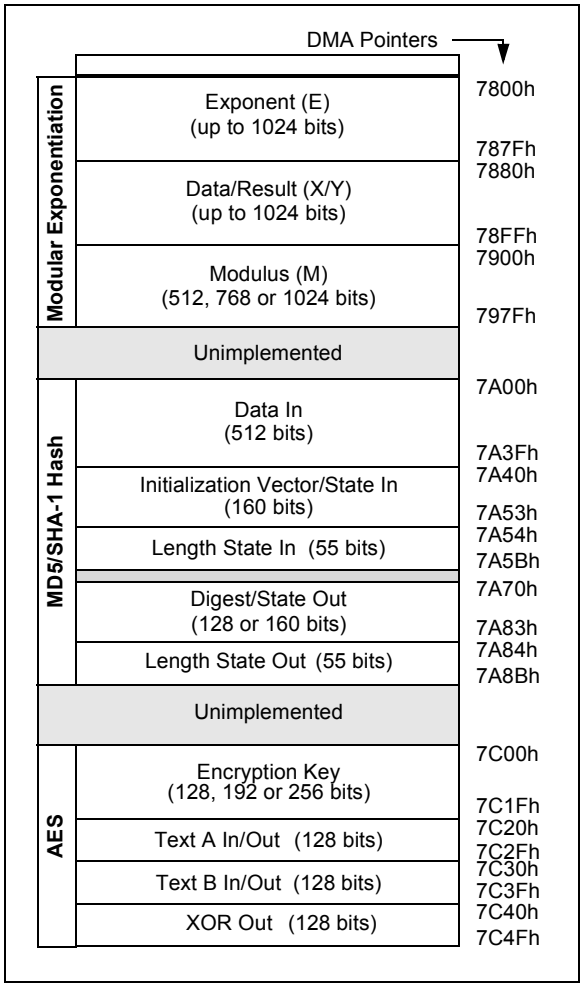
Legend: r = reserved bit, write as '0'; ignore on read; x = unknown. Reset values are shown in hexadecimal for each byte.

3.4 Cryptographic Data Memory

The cryptographic data memory is used to store key and data information for the Modular Exponentiation, AES and MD5/SHA-1 hashing engines. The RAM for these modules is actually implemented inside of the modules themselves; this allows fast memory access for the access-intensive encryption engines, as well as the simultaneous use of more than one module by an application. This memory is mapped into an area of address space that is accessible only by the DMA controller. The host controller must write to the cryptographic data memory by writing data to the 24-Kbyte SRAM buffer, then using the DMA to copy it into the security engine. Reading is performed in the opposite order, using the DMA to copy the data out of the security engine and into the SRAM buffer.

The mapping of the cryptographic space is shown in Figure 3-3. For additional information on the cryptographic engines, refer to **Section 15.0 "Cryptographic Security Engines"**. For additional information on the DMA controller, see **Section 14.0 "Direct Memory Access (DMA) Controller"**.

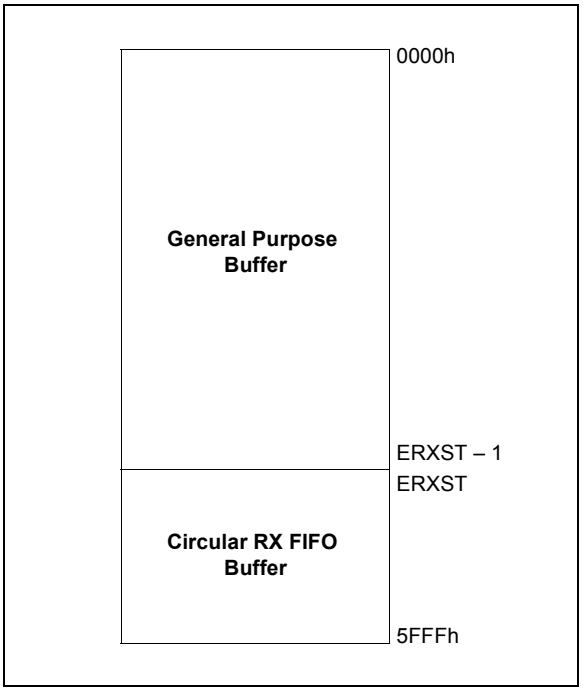
FIGURE 3-3: CRYPTOGRAPHIC DATA MEMORY MAPPING



3.5 SRAM Buffer

The SRAM buffer is a bulk 12K word x 16-bit (24 Kbytes) memory, used for TX/RX packet buffering and general purpose storage by the host microcontroller. In most cases, the memory is accessed using a byte-oriented interface, so the memory can normally be thought of as a simple 24-Kbyte memory buffer divided into a general purpose/TX area and an RX area (Figure 3-4).

FIGURE 3-4: SRAM BUFFER ORGANIZATION



Ethernet communications on 10Base-T and 100Base-TX networks occur at a fixed speed of 10 Mbps or 100 Mbps, respectively. Intra-byte gaps are not allowed. This requires the host controller to build outbound transmit frames in their entirety in the SRAM buffer before the hardware is allowed to begin transmission. Similarly, when receiving packets, the buffer provides space for the hardware to write the incoming packet without forcing the host microcontroller to immediately read and process the packet.

After the part exits Reset, the entire buffer is accessible by the host controller, regardless of other transmit, receive or DMA operations that may simultaneously also be accessing the general purpose or receive buffer memory.

3.5.1 GENERAL PURPOSE BUFFER

The general purpose buffer memory starts at address 0000h and includes all memory up to, but not including, the memory address pointed to by the ERXST register (i.e., ERXST – 1).

This buffer can be used to store transmit packets, received data that the host controller wishes to save for an extended period, or any type of volatile or state information that the host controller does not have room internally to save. Upper layer communications protocols and applications, such as a TCP/IP stack with SSL or TLS security, are generally infeasible or will perform poorly over high latency Internet links without using large buffers.

For reliable, connection oriented protocols like TCP, the maximum theoretical throughput is directly proportional to the round trip Acknowledgement latency of the link and the size of the corresponding transmit or receive buffer. The general purpose buffer memory on the ENCX24J600 is well suited for use by TCP for implementing high-performance communications across the Internet, where round trip Acknowledgement latency is in the order of many milliseconds.

3.5.2 RECEIVE BUFFER

The receive buffer constitutes a circular FIFO buffer managed by hardware. The buffer extends inclusively from the byte pointed to by the ERXST Pointer, to the very end of the SRAM at address 5FFFh. The size of the buffer, in bytes, is therefore defined as:

$$\text{RX Buffer Size} = 5FFFh - ERXST + 1$$

As bytes of data are received from the Ethernet interface, they are written into the receive buffer sequentially. However, after the memory at address 5FFFh is written to, the hardware will automatically wrap around and write the next byte of received data to the ERXST address. As a result, the receive hardware will never write outside the boundaries of the RX FIFO buffer.

For proper 16-bit word alignment, the ERXST Pointer is required to point to an even memory address. The Least Significant bit of this register is read-only and fixed as '0' to force even alignment. All other implemented bits in this register are read/write and can be programmed by software to point to any even address, from 0000h to 5FFEh.

The default value of ERXST on device Reset is 5340h. This allocates 21,312 bytes to the general purpose buffer and 3,264 bytes to the RX buffer. This RX buffer size is adequate to store at least two maximum length Ethernet frames, or any combination of numerous smaller packets.

The host controller may only program the ERXST Pointer when the receive logic is disabled. The pointer must not be modified while the receive logic is enabled by having RXEN (ECON1<0>) set.

The receive memory is always accessible to the RX hardware, regardless of transmit, DMA operations or host controller read/write operations, which may be accessing the SRAM simultaneously. The RX hardware will never drop a packet due to insufficient memory access bandwidth.

3.5.3 TRANSMIT BUFFER

The ENC624J600 family does not implement a dedicated transmit buffer. The transmit hardware has the flexibility of transmitting data starting at any memory address, including odd memory addresses which are off of a 16-bit word boundary. The host controller can transmit data from either the general purpose area or RX FIFO area of the entire 24 Kbytes of SRAM.

Because of the transmit flexibility, the host controller may store many prebuilt packets in the general purpose buffer for quick transmission. Alternatively, because the hardware can transmit data from the receive buffer, it is possible to quickly modify certain packet header fields and retransmit received packets without reading the entire packet contents into the host microcontroller. This feature may improve performance on certain proxy, gateway or echoing ("ping") applications.

The transmit hardware performs reads from the SRAM only; it never writes anything into the SRAM.

The entire SRAM is always accessible to the TX hardware, regardless of the receive activity, DMA operations or host controller read/write operations, which may be simultaneously attempting to access the SRAM.

3.5.4 DIRECT SRAM BUFFER ACCESS

When one of the PSP interfaces is used, the SRAM buffer is directly accessible through the interface. Assuming that all necessary address lines are connected, all addresses in the memory maps shown in Figure 3-2 (except for the cryptographic data memory) may be directly read and written to. When accessed through this manner, the host controller must handle all address increment and wrap-around calculations that may be necessary. This also includes translation from byte to word addressing when a 16-bit PSP interface is used.

Direct access is unavailable when the SPI interface is used.

3.5.5 INDIRECT SRAM BUFFER ACCESS

Indirect access to the SRAM buffer is available to all I/O interfaces. For the SPI interface, it is the only method available. For PSP interfaces, it may be used in addition to the direct access method.

Three separate pointer pairs are available for the host microcontroller to use when accessing the SRAM:

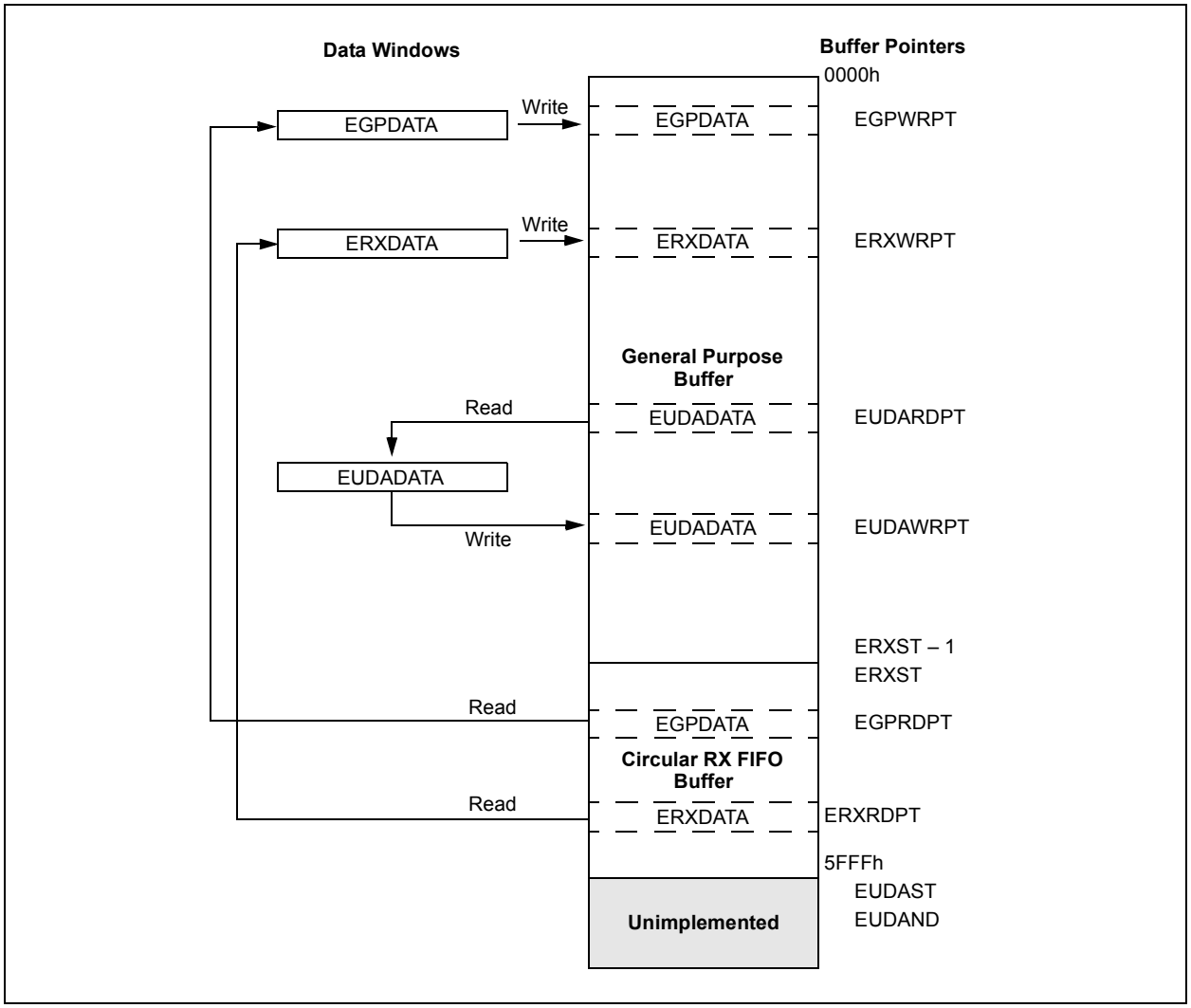
- General Purpose Buffer Read/Write Pointer (EGPRDPT/EGPWRPT)
- Receive Buffer Read/Write Pointer (ERXRDPT/ERXWRPT)
- User-Defined Area Read/Write Pointer (EUDARDPT/EUDAWRPT)

Each of these pointer pairs provides an 8-bit virtual window register (EGPDATA, ERXDATA and EUDATA) through which the SRAM data is read or written. The pointers and their associated data windows are shown in Figure 3-5.

EGPDATA, ERXDATA and EUDADATA are all 8 bits wide. When writing to them using a 16-bit PSP interface, the low-order byte select or write enable must be used; strobing the high byte Byte Select or Write Enable has no effect. When reading from a 16-bit PSP interface, one byte of useful data will be returned on the lower 8 bits of the data bus; the upper 8 bits are to be ignored.

When a data window register is read, the memory contents at the address indicated by the corresponding Read Pointer are obtained and presented to the host microcontroller. Similarly, when a data window register is written, the memory contents at the address indicated by the corresponding Write Pointer are updated by the data from the host microcontroller. Following a read/write operation, the appropriate pointer is automatically incremented in hardware.

FIGURE 3-5: POINTERS FOR INDIRECT BUFFER ACCESS



For example, to read data from address 5402h of the buffer:

1. Write 5402h to EGPRDPT.
2. Read from EGPDATA.

Following the read, the EGPRDPT value normally increments by 1 (to 5403h in this example). If the host subsequently wants to read from address 5403h, it can simply perform a second read from the EGPDATA Window register. The Write Pointer, EGPWRPT, is not affected by the read operation.

Similarly, to write A3h to address 0007h of the buffer:

1. Write 0007h to EGPWRPT.
2. Write A3h to EGPDATA.

Following the write, the EGPWRPT value normally increments by 1 (to 0008h in this example). The Read Pointer, EGPRDPT, is not affected by the write operation.

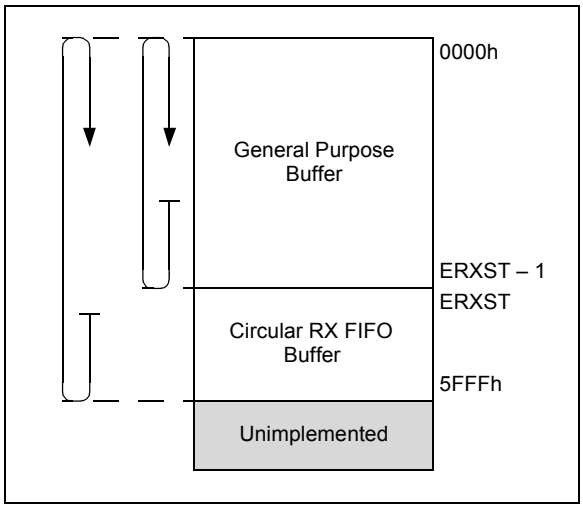
Each of the three pointer sets (general purpose, receive and user-defined area) can be used to access any address within the SRAM buffer. They differ from each other based on their address wrapping behavior.

Applications may choose to use all three pointer interfaces to access the RAM. This may offer maximum application performance as it will require minimal context switching overhead when, for example, switching from reading a received packet to reading from general purpose RAM. However, for simplicity, some applications may prefer to use only one or two of the three E*DATA interfaces.

3.5.5.1 Circular Wrapping with EGPDATA

Normally, operations involving EGPDATA cause the EGPRDPT or EGPWRPT Pointer to automatically increment by one byte address. However, if the end of the general purpose buffer area (ERXST – 1) is reached, or the end of the implemented SRAM (5FFFh) is reached, the pointer will increment to address 0000h instead, causing subsequent accesses to wrap around to the beginning of the SRAM buffer (Figure 3-6). The increment behavior logic is explained in Equation 3-1.

FIGURE 3-6: CIRCULAR BUFFER WRAPPING USING THE EGPDATA WINDOW



EQUATION 3-1: POINTER INCREMENT LOGIC FOR EGPRDPT AND EGPWRPT

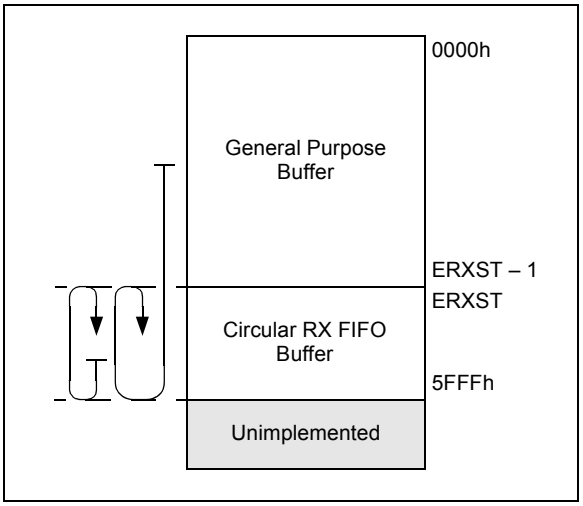
```
if EGPRDPT/EGPWRPT = ERXST – 1, then
        EGPRDPT/EGPWRPT = 0000h
else if EGPRDPT/EGPWRPT = 5FFFh, then
        EGPRDPT/EGPWRPT = 0000h
else
        EGPRDPT/EGPWRPT = EGPRDPT/EGPWRPT + 1
```

3.5.5.2 Circular Wrapping with ERXDATA

As with the general purpose pointers, operations with ERXDATA normally cause the ERXRDPT or ERXWRPT Pointer to automatically increment by one byte address. However, if the end of the receive buffer area (5FFFh) is reached, the pointer will increment to the start of the receive FIFO buffer area instead, as defined by ERXST (Figure 3-7).

The receive wrapping rules for the ERXDATA interface are identical to the buffer wrapping rules used by the receive hardware. Therefore, this register interface is ideally suited to reading packet data from the receive buffer. The host controller can set the ERXRDPT value at the start of a packet in the receive buffer and sequentially read out the entire packet contents without having to write to the ERXRDPT Read Pointer again.

FIGURE 3-7: CIRCULAR BUFFER WRAPPING USING THE ERXDATA WINDOW



EQUATION 3-2: POINTER INCREMENT LOGIC FOR ERXRDPT AND ERXWRPT

```
if ERXRDPT/ERXWRPT = 5FFFh, then
        ERXRDPT/ERXWRPT = ERXST
else
        ERXRDPT/ERXWRPT = ERXRDPT/ERXWRPT + 1
```

3.5.5.3 Circular Wrapping with EUDADATA

The user-defined buffer area is primarily useful for setting up a circular FIFO within the general purpose area for use by TCP/IP stacks or other applications. The wrap-around behavior of the user-defined buffer area is somewhat more complicated than with the general purpose or receive buffer cases. This is because the user-definable boundaries set by EUDAST and EUDAND take priority over normal wrapping behavior.

Like other pointers, EUDAST and EUDAND are fully user-configurable from the host microcontroller. Unlike ERXST, which must not be modified while the receive hardware is enabled, EUDAST and EUDAND can be modified at any time.

As in the previous instances, operations with EUDADATA normally cause the EUDARDPT or EUDAWRPT Pointer to automatically increment by one byte address. If the value in EUDAND is reached, the pointer will increment to the address specified by EUDAST instead. However, if the end of memory (5FFFh) is reached, and EUDAND is located at some other address, the pointer will increment to the beginning of memory (0000h). If EUDAND is set to 5FFFh, the pointer address increments to the value of EUDAST, instead of 0000h.

The increment behavior logic is explained in Equation 3-3.

EQUATION 3-3: POINTER INCREMENT LOGIC FOR EUDARDPT AND EUDAWRPT

```
if EUDARDPT/EUDAWRPT = EUDAND, then
        EUDARDPT/EUDAWRPT = EUDAST
else if EUDARDPT/EUDAWRPT = 5FFFh, then
        EUDARDPT/EUDAWRPT = 0000h
else
        EUDARDPT/EUDAWRPT = EUDARDPT/EUDAWRPT + 1
```

The user-defined area start address, EUDAST, is a read/write register. For wrapping to work correctly, the hardware enforces 16-bit even word alignment of this register by internally having the Least Significant bit tied off to '0'. Similarly, the user-defined area end address, EUDAND, is a read/write register that is forced to an odd memory address. The Least Significant bit of EUDAND is internally tied to '1'. Applications wishing to set up general purpose circular FIFOs in memory using these hardware features must observe these same alignment requirements.

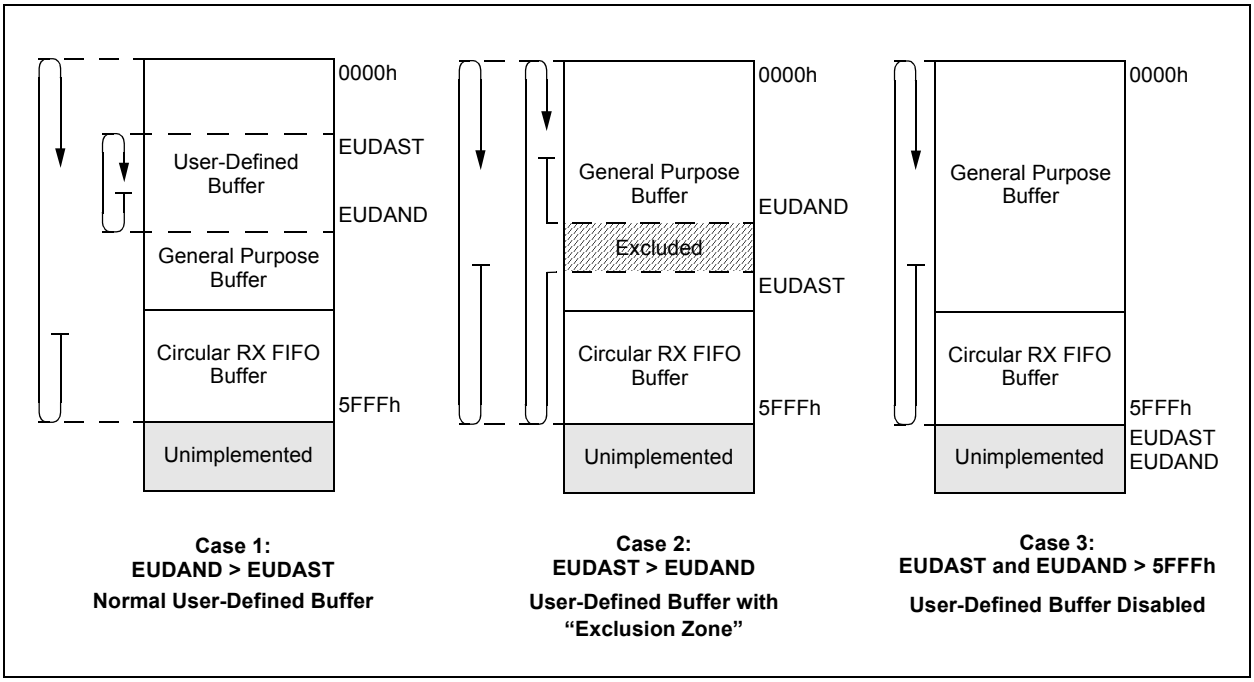
If the user-defined area end address, EUDAND, is at a higher memory address relative to the start address, EUDAST, the buffer wraps to either EUDAST or the beginning of memory, depending on where the EUDARDPT or EUDAWRPT Pointers are located. This is shown in Case 1 of Figure 3-8.

In some cases (for example, when accessing fragmented data), it may be useful to place the EUDAST Pointer at a higher memory address relative to the end address. When organized in such a manner, an "exclusion zone" in the middle of the memory range is created; sequential read/write operations with the user-defined area pointers will jump over the range of addresses between EUDAND and EUDAST. This is shown in Case 2.

If the user-defined buffer is not needed, it can effectively be disabled by setting EUDAST and EUDAND to addresses outside of the implemented memory area. For example, if EUDAST is set to 6000h and EUDAND is set to 6001h, EUDARDPT and EUDAWRPT will never reach these addresses. Instead, they wrap from the end of implemented RAM to its beginning, as shown in Case 3.

When the user-defined buffer is disabled, the host controller can use the EUDADATA interface as a second general purpose window into RAM. Unlike the original general purpose pointers, however, EUDARDPT and EUDAWRPT do not wrap at the ERXST boundary, thereby allowing access to the full SRAM buffer area. This may be beneficial for debugging and testing purposes where it may be desirable to read or write the entire SRAM buffer in a single operation.

FIGURE 3-8: CIRCULAR BUFFER WRAPPING USING THE EUDATA WINDOW



NOTES:

4.0 SERIAL PERIPHERAL INTERFACE (SPI)

ENC424J600/624J600 devices implement an optional SPI I/O port for applications where a parallel microcontroller interface is not available or is undesirable. An SPI port is commonly available on many microcontrollers, and can be simulated in software on regular I/O pins where it is not implemented. This makes the SPI port ideal for using ENC424J600/624J600 devices with the widest possible range of host controllers.

4.1 Physical Implementation

The SPI port on ENC424J600/624J600 devices operates as a slave port only. The host controller must be configured as an SPI master that generates the Serial Clock (SCK) signal.

This implementation supports SPI Mode 0,0, which requires:

- SCK is Idle at a logic low state
- Data is clocked in on rising clock edges and changes on falling clock edges

Other SPI modes that use inverted clock polarity and/or phase are not supported.

Commands and data are sent to the device on the SI pin. Data is driven out on the SO line on the falling edge of SCK. The $\overline{CS}$ pin must be held low while any operation is performed, and returned to logic high when finished.

When $\overline{CS}$ is in the inactive (logic high) state, the SO pin is set to a high-impedance state and becomes 5V tolerant. This allows the ENCX24J600 to be connected to a single SPI bus shared by multiple SPI slave devices that also go to a high-impedance state when inactive.

For details on the physical connections to the interface, see **Section 2.7 "Host Interface Pins"**.

4.2 SPI Instruction Set

The SPI interface supports a unique instruction set, consisting of 47 distinct opcodes. These include a large number of optimized opcodes that perform a wide range of frequently performed operations with a minimum of SPI protocol overhead. Complete Ethernet functionality can be achieved with as few as six N-byte opcodes. The use of the other 41 is optional; however, doing so will generally improve overall system performance.

The SPI opcodes are divided into four families:

- Single Byte: Direct opcode instructions; designed for task-oriented SFR operations with no data returned
- Two-Byte: Direct opcode instruction; designed for SFR operation with byte data returned
- Three-Byte: Opcode with word length argument; includes read and write operations, designed for pointer manipulation with word length data returned
- N-Byte: Opcode with one or more bytes of argument; includes read and write operations designed for general memory space access with one or more bytes of data returned

A complete summary of all opcodes is provided in Table 4-1. A detailed explanation of each opcode family follows.

TABLE 4-1: SPI INSTRUCTION SET

Instruction	Mnemonic	Instruction			
		1st Byte	2nd Byte	3rd Byte	Nth Byte
Bank 0 Select	B0SEL	1100 0000	—	—	—
Bank 1 Select	B1SEL	1100 0010	—	—	—
Bank 2 Select	B2SEL	1100 0100	—	—	—
Bank 3 Select	B3SEL	1100 0110	—	—	—
System Reset	SETETHRST	1100 1010	—	—	—
Flow Control Disable	FCDISABLE	1110 0000	—	—	—
Flow Control Single	FCSINGLE	1110 0010	—	—	—
Flow Control Multiple	FCMULTIPLE	1110 0100	—	—	—
Flow Control Clear	FCCLEAR	1110 0110	—	—	—
Decrement Packet Counter	SETPKTDEC	1100 1100	—	—	—
DMA Stop	DMASTOP	1101 0010	—	—	—
DMA Start Checksum	DMACKSUM	1101 1000	—	—	—
DMA Start Checksum with Seed	DMACKSUMS	1101 1010	—	—	—
DMA Start Copy	DMACOPY	1101 1100	—	—	—
DMA Start Copy and Checksum with Seed	DMACOPYS	1101 1110	—	—	—
Request Packet Transmission	SETTXRTS	1101 0100	—	—	—
Enable RX	ENABLERX	1110 1000	—	—	—
Disable RX	DISABLERX	1110 1010	—	—	—
Enable Interrupts	SETEIE	1110 1100	—	—	—
Disable Interrupts	CLREIE	1110 1110	—	—	—
Read Bank Select	RBSEL	1100 1000	xxxx xxxx	—	—
Write EGPRDPT	WGPRDPT	0110 0000	dddd dddd	DDDD DDDD	—
Read EGPRDPT	RGPRDPT	0110 0010	xxxx xxxx	XXXX XXXX	—
Write ERXRDPT	WRXRDPT	0110 0100	dddd dddd	DDDD DDDD	—
Read ERXRDPT	RRXRDPT	0110 0110	xxxx xxxx	XXXX XXXX	—
Write EUDARDPT	WUDARDPT	0110 1000	dddd dddd	DDDD DDDD	—
Read EUDARDPT	RUDARDPT	0110 1010	xxxx xxxx	XXXX XXXX	—
Write EGPWRPT	WGPWRPT	0110 1100	dddd dddd	DDDD DDDD	—
Read EGPWRPT	RGPWRPT	0110 1110	xxxx xxxx	XXXX XXXX	—
Write ERXWRPT	WRXWRPT	0111 0000	dddd dddd	DDDD DDDD	—
Read ERXWRPT	RRXWRPT	0111 0010	xxxx xxxx	XXXX XXXX	—
Write EUDAWRPT	WUDAWRPT	0111 0100	dddd dddd	DDDD DDDD	—
Read EUDAWRPT	RUDAWRPT	0111 0110	xxxx xxxx	XXXX XXXX	—
Read Control Register	RCR	000a aaaa	xxxx xxxx	XXXX XXXX	XXXX XXXX
Write Control Register	WCR	010a aaaa	dddd dddd	DDDD DDDD	DDDD DDDD
Read Control Register Unbanked	RCRU	0010 0000	AAAA AAAA	xxxx xxxx	XXXX XXXX
Write Control Register Unbanked	WCRU	0010 0010	AAAA AAAA	dddd dddd	DDDD DDDD
Bit Field Set	BFS	100a aaaa	dddd dddd	DDDD DDDD	DDDD DDDD
Bit Field Clear	BFC	101a aaaa	dddd dddd	DDDD DDDD	DDDD DDDD
Bit Field Set Unbanked	BFSU	0010 0100	AAAA AAAA	dddd dddd	DDDD DDDD
Bit Field Clear Unbanked	BFCU	0010 0110	AAAA AAAA	dddd dddd	DDDD DDDD
Read EGPDATA	RGPDATA	0010 1000	xxxx xxxx	XXXX XXXX	XXXX XXXX
Write EGPDATA	WGPDATA	0010 1010	dddd dddd	DDDD DDDD	DDDD DDDD
Read ERXDATA	RRXDATA	0010 1100	xxxx xxxx	XXXX XXXX	XXXX XXXX
Write ERXDATA	WRXDATA	0010 1110	dddd dddd	DDDD DDDD	DDDD DDDD
Read EUDADATA	RUDADATA	0011 0000	xxxx xxxx	XXXX XXXX	XXXX XXXX
Write EUDADATA	WUDADATA	0011 0010	dddd dddd	DDDD DDDD	DDDD DDDD

Legend: x/X = read data, d/D = write data, a = banked SFR address, A = unbanked SFR address. 'X' and 'D' are optional.

4.3 Single Byte Instructions

All single byte instructions are designed to perform a simple command that affects the ENCX24J600 device's state. In most cases, they set or clear a small number of control bits which would otherwise require one or more N-byte opcodes to perform. None of these instructions return any data to the host microcontroller.

Figure 4-1 shows the timing relationships for performing a single byte operation. The opcode ('11xxxxx0') is presented on the device's SI pin starting with the Most Significant bit of the opcode; the Least Significant bit is always '0'. The SO pin is actively driven with indeterminate '1's or '0's while the $\overline{CS}$ pin is driven low. It continues to be driven until the $\overline{CS}$ pin is returned high.

Because all single byte instructions are fixed length with no optional parameters, it is possible to execute any instruction immediately following the execution of any single byte instruction without deasserting the chip select line in between.

If the $\overline{CS}$ control signal is deactivated before the 8th bit of the opcode is sent to the ENCX24J600, indeterminate results will occur. In some cases, the instruction is executed or partially executed. To avoid this, it is recommended that a single byte instruction should not be interrupted. If it is unavoidable that an instruction gets partially executed, have the application later reissue the same instruction and let it complete to place the device into a known state.

There are a total of 20 single byte opcodes, which are listed in Table 4-2. All single byte opcodes will operate regardless of which SFR bank is selected at the time. Those opcodes that affect multiple bits, or affect SFR addressing, are detailed below.

4.3.1 BxSEL OPCODES

The bank select opcodes, B0SEL, B1SEL, B2SEL and B3SEL, switch the SFR bank to Bank 0, Bank 1, Bank 2 or Bank 3, respectively. The updated bank select state is saved internally inside the ENCX24J600 in volatile memory. Firmware can retrieve the currently selected SFR bank state by using the Read Bank Select (RBSEL) opcode.

The bank select opcodes are needed to access most SFR addresses when using the RCR, WCR, BFS and BFC instructions. These are discussed in more detail in **Section 4.6 "N-Byte Instructions"**.

Upon device power-up or System Reset, Bank 0 is automatically selected. After Reset, hardware does not modify the bank state again. Any value programmed by a BxSEL opcode is retained until the next BxSEL opcode is executed or a System Reset is issued.

4.3.2 FC (FLOW CONTROL) OPCODES

The flow control opcodes, FCDISABLE, FCSINGLE, FCMULTIPLE and FCCLEAR, all modify the device's Flow Control mode by changing the values of the FCOP<1:0> bits (ECON1<7:6>). These opcodes execute regardless of the currently selected SFR bank. For more information on flow control operation, see **Section 11.0 "Flow Control"**.

4.3.3 DMA OPCODES

The DMA opcodes, DMASTOP, DMACKSUM, DMACKSUMS, DMACOPY and DMACOPYS, modify the operation of the device's DMA controller, all by simultaneously changing the values of the DMAST, DMACPY, DMACSSD and DMANOCS control bits (ECON1<5:2>). For more information on DMA operation, see **Section 14.0 "Direct Memory Access (DMA) Controller"**.

FIGURE 4-1: SINGLE BYTE INSTRUCTION TIMING

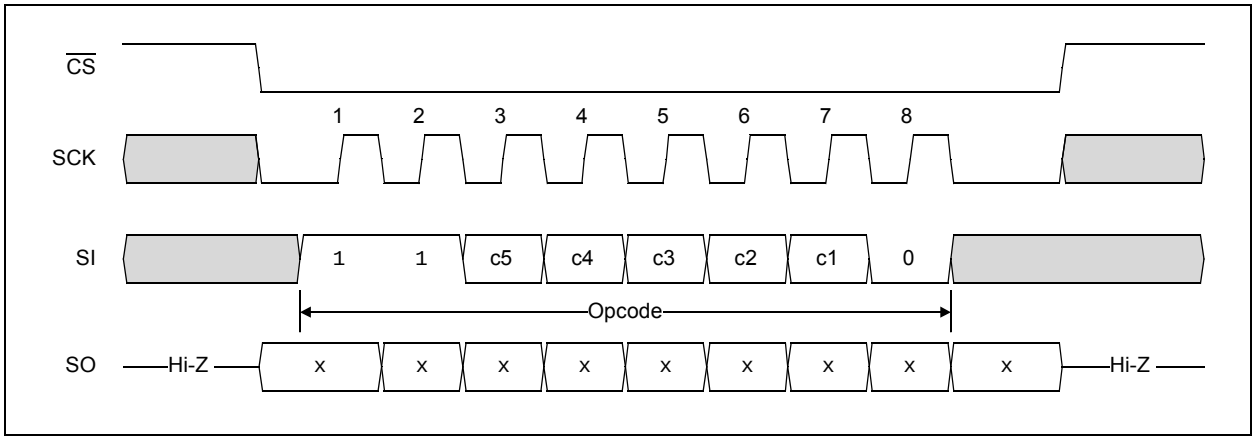


TABLE 4-2: SINGLE BYTE INSTRUCTIONS

Mnemonic	Opcode	Instruction
B0SEL	1100 0000	Selects SFR Bank 0
B1SEL	1100 0010	Selects SFR Bank 1
B2SEL	1100 0100	Selects SFR Bank 2
B3SEL	1100 0110	Selects SFR Bank 3
SETETHRST	1100 1010	Issues System Reset by setting ETHRST (ECON2<4>)
FCDISABLE	1110 0000	Disables flow control (sets ECON1<7:6> = 00)
FCSINGLE	1110 0010	Transmits a single pause frame (sets ECON1<7:6> = 01)
FCMULTIPLE	1110 0100	Enables flow control with periodic pause frames (sets ECON1<7:6> = 10)
FCCLEAR	1110 0110	Terminates flow control with a final pause frame (sets ECON1<7:6> = 11)
SETPKTDEC	1100 1100	Decrements PKTCNT by setting PKTDEC (ECON1<8>)
DMASTOP	1101 0010	Stops current DMA operation by clearing DMAST (ECON1<5>)
DMACKSUM	1101 1000	Starts DMA and checksum operation (sets ECON1<5:2> = 1000)
DMACKSUMS	1101 1010	Starts DMA checksum operation with seed (sets ECON1<5:2> = 1010)
DMACOPY	1101 1100	Starts DMA copy and checksum operation (sets ECON1<5:2> = 1100)
DMACOPYS	1101 1110	Starts DMA copy and checksum operation with seed (sets ECON1<5:2> = 1110)
SETTXRTS	1101 0100	Sets TXRTS (ECON1<1>), sends an Ethernet packet
ENABLERX	1110 1000	Enables packet reception by setting RXEN (ECON1<0>)
DISABLERX	1110 1010	Disables packet reception by clearing RXEN (ECON1<0>)
SETEIE	1110 1100	Enable Ethernet Interrupts by setting INT (ESTAT<15>)
CLREIE	1110 1110	Disable Ethernet Interrupts by clearing INT (ESTAT<15>)

4.4 Two-Byte Instructions

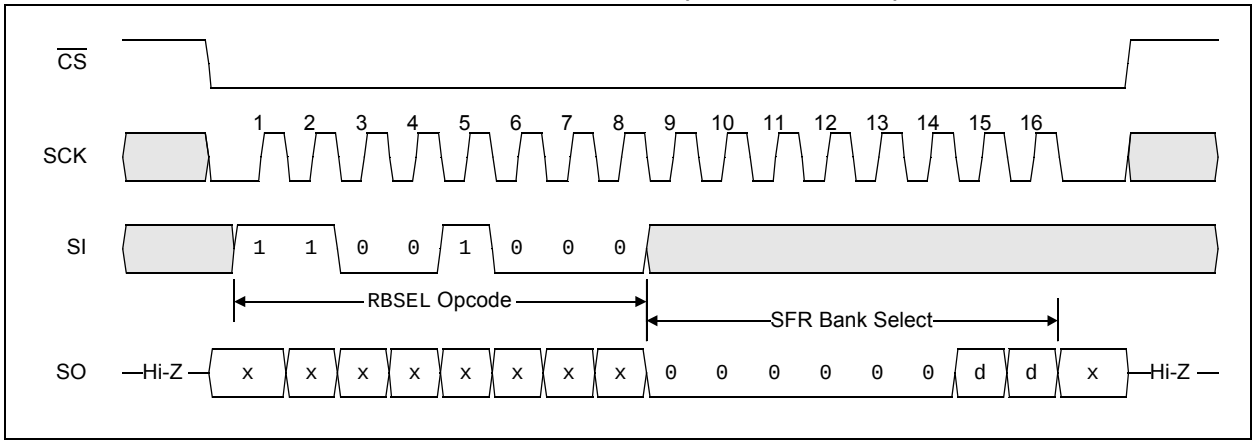
There is only one instruction in the ENCX24J600 command set which uses two SPI bytes. The Read Bank Select opcode, RBSEL, reads the internal SFR bank select state and returns the value to the host controller.

Figure 4-2 shows the timing relationships for performing the two-byte operation. The first byte of the opcode (‘11001000’) must be presented on the SI pin, MSb first, followed by “don’t care” values for the second byte (9th through 16th SCK rising edges). The bank select value (00h through 03h) is returned on the SO pin, MSb first, while the second byte is being presented on the SI pin.

Because this instruction is a fixed length with no optional parameters, it is possible to execute any instruction following the execution of RBSEL without deasserting the chip select line in between.

Since this opcode does not modify the ENCX24J600 internal state, it can be aborted at any time by returning the $\overline{CS}$ pin to the inactive state.

FIGURE 4-2: TWO-BYTE INSTRUCTION TIMING (RBSEL OPCODE)



4.5 Three-Byte Instructions

All three-byte instructions are designed to quickly read or update the Read and Write Pointers used to access the SRAM buffer area. Unlike the single byte instructions and RBSEL, each instruction in this group has distinct read and write implementations.

For read commands (shown in Figure 4-3), the opcode byte ('011xxx10') must be presented on the SI pin, MSb first, followed by "don't care" values for the second and third bytes (9th through 24th SCK rising edges). Response data is returned on the SO line during the second and third bytes.

Data on the SO line is also presented in MSb first bit ordering. However, read commands are intended to read a 16-bit pointer in little-endian byte ordering. Therefore, the first byte on the SO line (returned during SCK clocks, 9 through 16) is the lower byte of the 16-bit pointer and is followed by the upper byte (returned during SCK clocks 17 through 24).

Read operations do not affect the ENCX24J600 device's internal state, and therefore, can be aborted at any time by deasserting chip select.

For write commands (shown in Figure 4-4), the opcode byte ('011xxx00') must be presented on the SI line, MSb first, followed immediately by the pointer data to be written. Like the data returned during a read operation, the write data must be presented MSb first, Least Significant Byte first.

If the application only needs to write to the lower byte of a 16-bit pointer, it can optionally skip the upper byte by raising chip select after the 16th clock pulse and allowing adequate chip select hold time to elapse. The hardware would then update the lower byte of the pointer while maintaining the original value in the upper byte.

During write operations, the device actively drives the SO line while the chip select line is active. The value during this interval is to be ignored.

All three-byte instructions, including read operations, are considered to be finished at the end of the 24th SCK clock (if reached). The host controller may issue another SPI instruction or multiple fixed length instructions without deasserting chip select.

There are 12 three-byte instructions, which are divided equally between read and write instructions. They are listed in Table 4-3.

FIGURE 4-3: THREE-BYTE READ INSTRUCTION TIMING

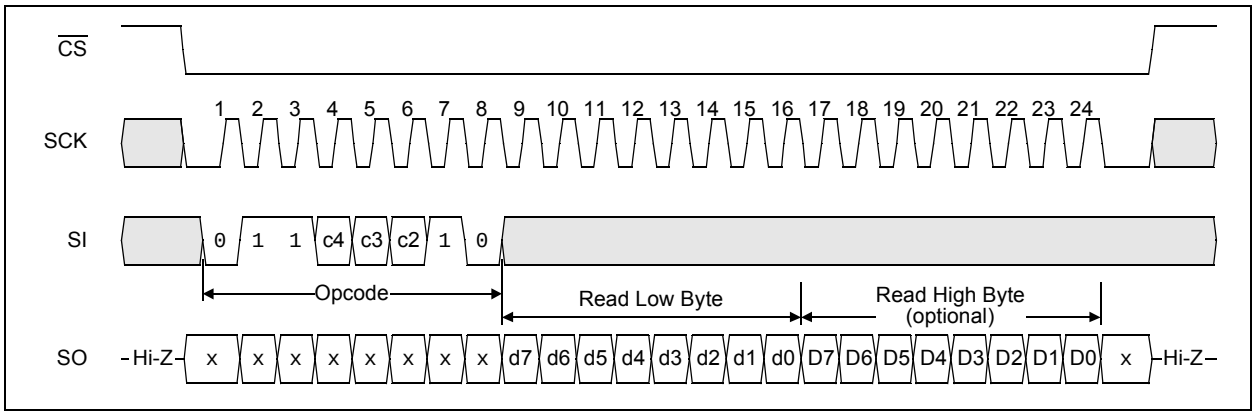


FIGURE 4-4: THREE-BYTE WRITE INSTRUCTION TIMING

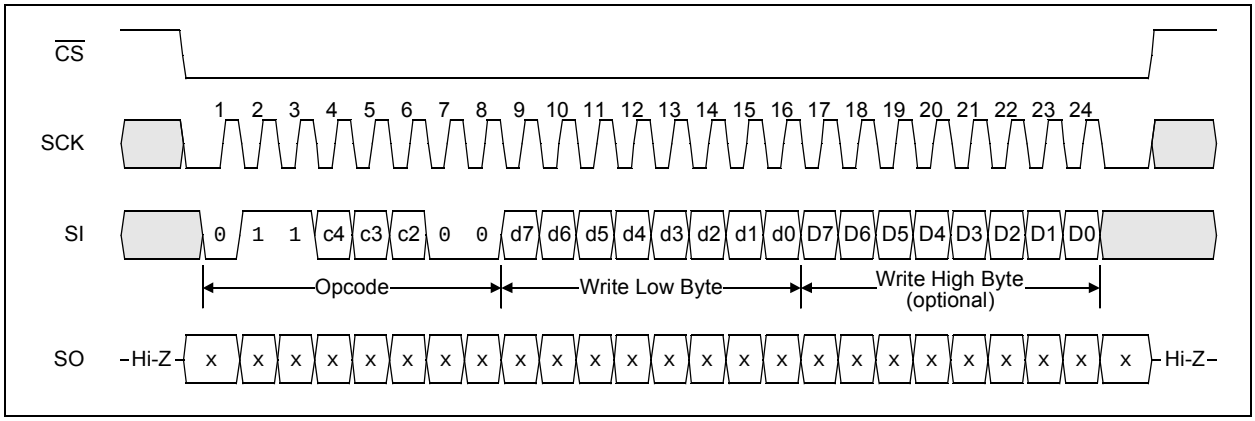


TABLE 4-3: THREE-BYTE INSTRUCTIONS

Mnemonic	Opcode	Argument		Instruction
	1st Byte	2nd Byte	3rd Byte	
WGPRDPT	0110 0000	dddd dddd	DDDD DDDD	Write General Purpose Buffer Read Pointer (EGPRDPT).
RGPRDPT	0110 0010	xxxx xxxx	XXXX XXXX	Read General Purpose Buffer Read Pointer (EGPRDPT).
WRXRDPT	0110 0100	dddd dddd	DDDD DDDD	Write Receive Buffer Read Pointer (ERXRDPT).
RRXRDPT	0110 0110	xxxx xxxx	XXXX XXXX	Read Receive Buffer Read Pointer (ERXRDPT).
WUDARDPT	0110 1000	dddd dddd	DDDD DDDD	Write User-Defined Area Read Pointer (EUDARDPT).
RUDARDPT	0110 1010	xxxx xxxx	XXXX XXXX	Read User-Defined Area Read Pointer (EUDARDPT).
WGPWRPT	0110 1100	dddd dddd	DDDD DDDD	Write General Purpose Buffer Write Pointer (EGPWRPT).
RGPWRPT	0110 1110	xxxx xxxx	XXXX XXXX	Read General Purpose Buffer Write Pointer (EGPWRPT).
WRXWRPT	0111 0000	dddd dddd	DDDD DDDD	Write Receive Buffer Write Pointer (ERXWRPT).
RRXWRPT	0111 0010	xxxx xxxx	XXXX XXXX	Read Receive Buffer Write Pointer (ERXWRPT).
WUDAWRPT	0111 0100	dddd dddd	DDDD DDDD	Write User-Defined Area Write Pointer (EUDAWRPT).
RUDAWRPT	0111 0110	xxxx xxxx	XXXX XXXX	Read User-Defined Area Write Pointer (EUDAWRPT).

Legend: x/d = pointer data (LSB), X/D = pointer data (MSB, optional).

4.6 N-Byte Instructions

N-byte instructions make up the most versatile class of SPI commands, as they can read or write to any addressable SFR or SRAM space. Their name comes from their variable length nature; they require a minimum of two bytes, but can take an indefinite number of bytes of data argument, or return an unlimited number of output bytes. This makes them useful for reading or writing entire arrays of data to or from the SRAM buffer.

Since these instructions are of an intrinsically variable length, no other opcode may follow any N-byte instruction until the $\overline{CS}$ line is driven high. Driving $\overline{CS}$ high terminates the instruction and then places the SO pin in a high-impedance state.

The format of the N-byte instructions differs depending on if a read versus a write command is executed, and if a banked SFR, unbanked SFR or SRAM location is accessed. The differences are discussed in the following sections.

4.6.1 BANKED SFR OPERATION

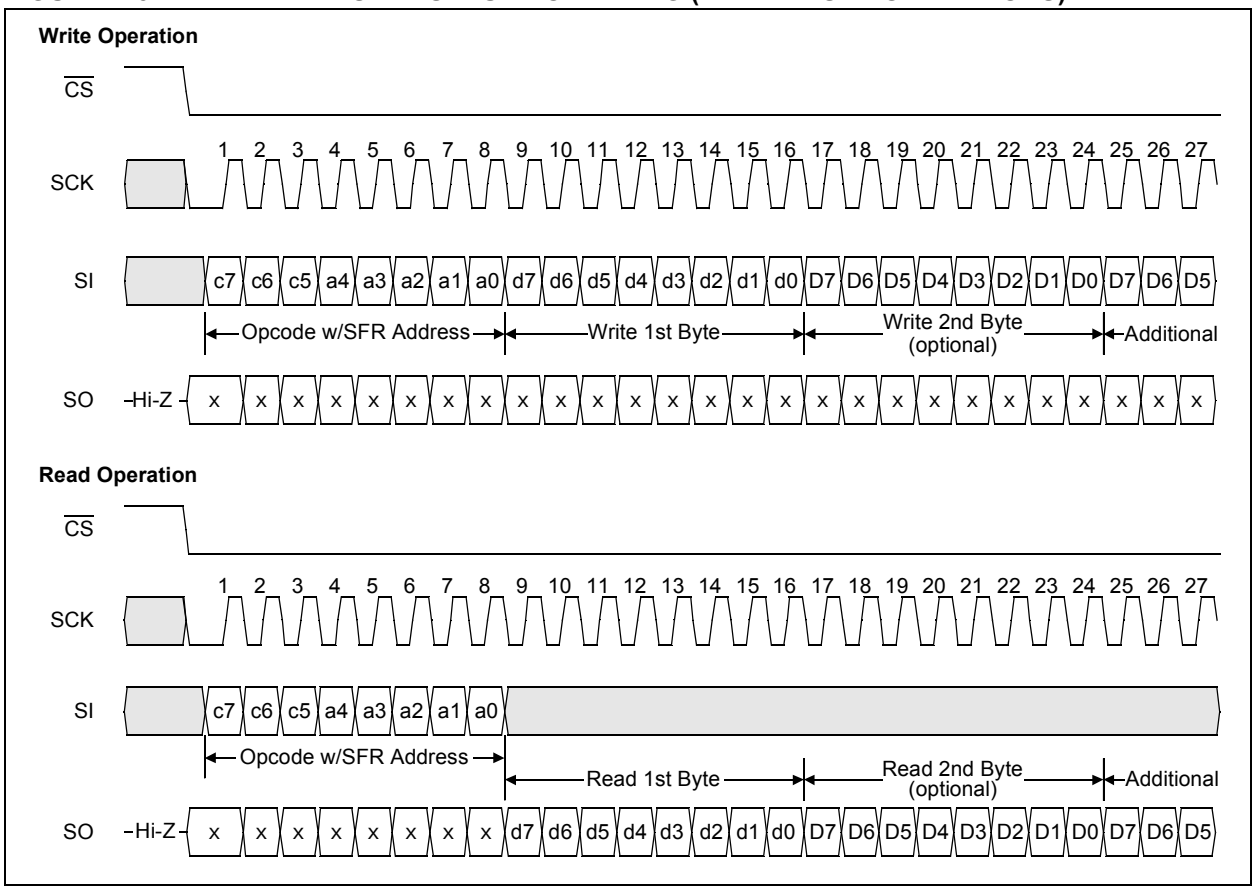
The N-byte Banked SFR instructions are WCR, RCR, BFS and BFC. These instructions depend on the use of the appropriate BxSEL instructions to select the proper SFR bank prior to their execution. Because of this, they cannot be used for the unbanked SFR space (80h through 9Fh).

Figure 4-5 shows the timing relationships for these operations. Like all other opcodes, data must be presented on the SI pin, MSb first. For all banked instructions, the first byte of data must be the opcode, comprised of a 3-bit prefix designating the instruction and a 5-bit banked SFR address. If the instruction is a write or bit field set/clear opcode, the next bytes are the data or bit mask to be written. For read instructions, the next bytes on the SI pin are “don’t care”.

For write and bit field set/clear instructions, the SO pin is actively driven with indeterminate ‘1’s or ‘0’s while the $\overline{CS}$ pin is driven low. For read instructions, indeterminate data is clocked out on SO during SCK clocks, 1 through 8. Starting with the 9th clock, valid data is clocked out byte-wise on SO, MSb first.

As long as the $\overline{CS}$ pin is held low, clocks on SCK are provided and data is presented on SI, the instruction continues to execute indefinitely, automatically incrementing to the next register address in the SFR Bank and writing data from SI to, or outputting data on SO from, subsequent registers. When the end of a bank is reached, the address automatically wraps back to the beginning (00h) of the bank and continues; the selected bank does not change.

FIGURE 4-5: N-BYTE SPI INSTRUCTION TIMING (BANKED SFR OPERATIONS)



There are four banked SFR opcodes, summarized in Table 4-4. Additional details for these opcodes are provided below.

4.6.1.1 WCR Opcode

The Write Control Register (WCR) opcode byte consists of the prefix, '010', concatenated with the 5-bit banked SFR address of the first register to write to. For example, if Bank 3 were currently selected and the host microcontroller wanted to write to the ECON2L register at banked address 0Eh, the 8-bit opcode would be '01001110' or 4Eh.

Generally, WCR can be executed on most register addresses, in any sequence and for any length. An important exception is when WCR is used on any MAC or MII register. These registers must be written as a whole 16-bit register, low byte first (e.g., MACON1 must be written by first writing to MACON1L, then MACON1H). Writing only to the upper byte of a MAC or MII register results in a successful write to the upper register, while the lower register is written with indeterminate data. If a WCR instruction is aborted by raising $\overline{CS}$ while writing to the upper byte of a MAC or MII register, neither upper nor lower byte will be updated.

4.6.1.2 RCR Opcode

The Read Control Register (RCR) opcode byte consists of the prefix, '000', concatenated with the 5-bit banked SFR address of the first register to read from. Using the previous example, the 8-bit opcode to read ECON2L would be '00001110' or 0Eh.

Read operations can be performed against any register address, in any sequence and for any length. However, due to volatile register shadowing, it is recommended that the ERXHEADH:ERXHEADL register pair be read in sequence (low byte first) to obtain the correct value. See **Section 9.2 "Receiving Packets"** for additional information.

4.6.1.3 BFS and BFC Opcodes

The Bit Field Set (BFS) and Bit Field Clear (BFC) opcodes consist of the prefix, '100' (for BFS) or '101' (for BFC), concatenated with the 5-bit banked SFR address of the first register to write to. In terms of timing and automatic address increment, they behave almost identically to the WCR opcode. However, instead of absolute data to be written to a register, the host microcontroller provides a bit mask showing which bits of the target register need to be set or cleared.

For BFS, the ENCX24J600 performs a logical OR operation with the supplied bit field causing '1' bits in the bit field to become set bits in the register; '0' bits in the bit field have no effect on the corresponding register bits. For BFC, the ENCX24J600 performs a logical AND with the *complement* of the supplied mask. This causes '1' bits in the mask to become clear bits in the register; '0' bits in the mask do not affect the corresponding register bits.

The host controller must use bit field operations when attempting to change bits in a volatile control or interrupt flag register. Normally, changing such a bit might be accomplished by the application as a "read-modify-write" operation: reading the control register's contents, modifying the register copy in memory on the controller side and writing the modified register data back to the ENCX24J600. In a dynamic environment, however, one or more control bits may change state between the read and write, resulting in an incorrect device state after the write. As an example, assume that the DMA module is in use (ECON1L<5> = 1) at the same time that the application wants to transmit a packet (i.e., setting ECON1L<1>). By the time a read-modify-write on ECON1L is complete, the DMA operation may have completed and cleared ECON1L<5>. In this case, the write back erroneously starts a new DMA operation.

Using BFS and BFC allows for bit level changes to one or more control bits without the delay of a read and write back. In the previous example, using BFS with a bit mask of '00000010' on ECON1L, sets ECON1L<1> and starts a packet transmission without affecting the status of ECON1L<5>.

Note:	Unlike the WCR opcode, BFS and BFC cannot be used to modify MAC or MII registers. Never use these opcodes on MAC and MII registers.

TABLE 4-4: N-BYTE BANKED SFR INSTRUCTIONS

Instruction	Mnemonic	Opcode	Argument		
		1st Byte	2nd Byte	3rd Byte	Nth Byte
Read Control Register(s)	RCR	000a aaaa	xxxx xxxx	XXXX XXXX	XXXX XXXX
Write Control Register(s)	WCR	010a aaaa	dddd dddd	DDDD DDDD	DDDD DDDD
Bit Field(s) Set	BFS	100a aaaa	dddd dddd	DDDD DDDD	DDDD DDDD
Bit Field(s) Clear	BFC	101a aaaa	dddd dddd	DDDD DDDD	DDDD DDDD

Legend: x/X = read data (LSB/MSB), d/D = write data (LSB/MSB), a = banked SFR address.

4.6.2 UNBANKED SFR OPERATIONS

The N-byte unbanked SFR instructions are WCRU, RCRU, BFSU and BFCU. These instructions use an opcode with a one-byte address argument and do not depend on the use of BxSEL instructions for SFR bank selection.

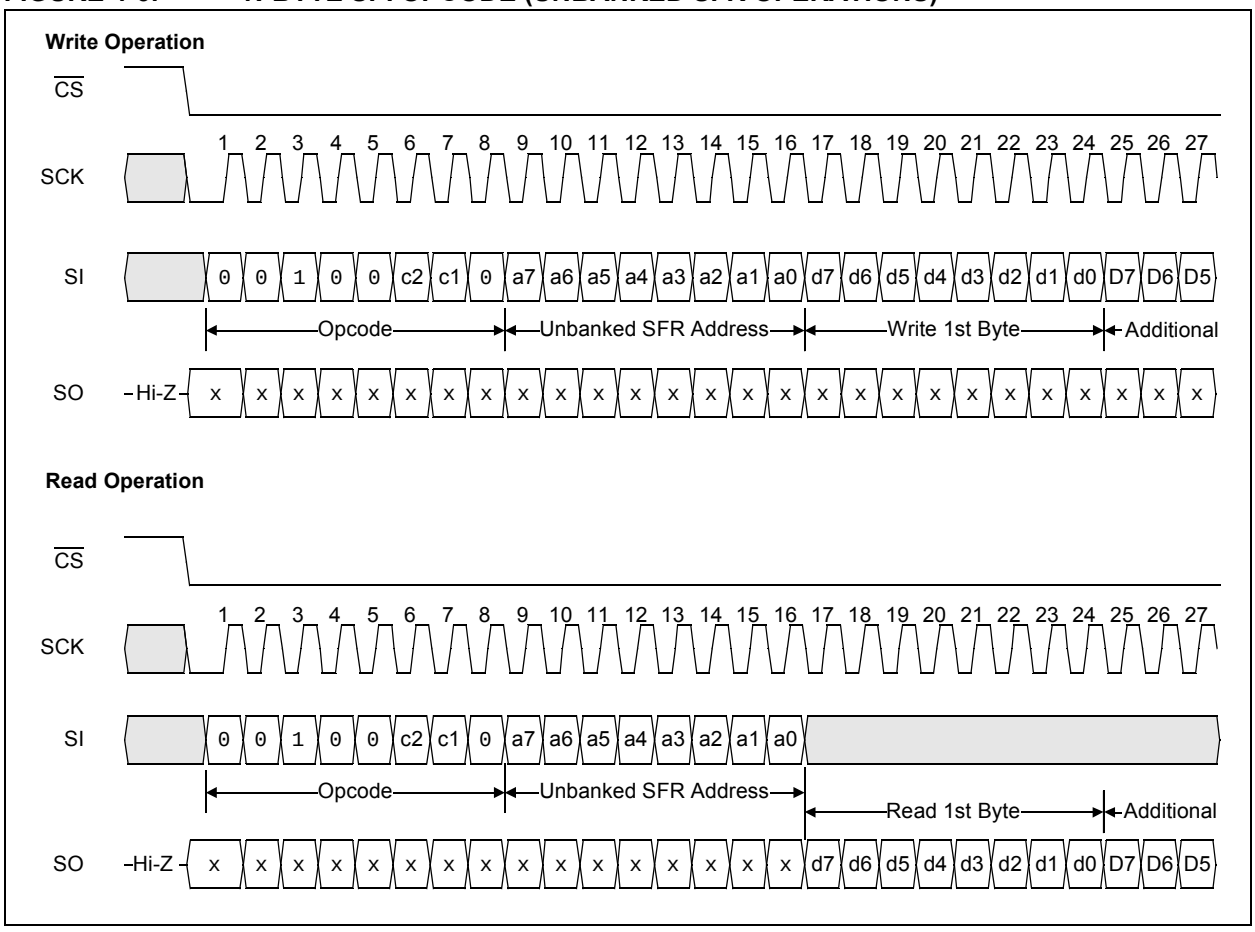
Figure 4-6 shows the timing relationships for these operations. Like all other opcodes, data is presented on the SI pin, MSb first. For this class of instructions, the first byte of data is a specific opcode; the second byte is the 8-bit absolute address of the target SFR. If the instruction is a write or bit set/clear opcode, the next bytes are the data or bit mask to be written. For read instructions, the next bytes are don't cares.

For write and bit set/clear instructions, the SO pin is actively driven with indeterminate '1's or '0's while the $\overline{CS}$ pin is driven low. For read instructions, random data is clocked out on SO during SCK clocks, 1 through 16. Starting with the 17th clock, data is clocked out byte-wise on SO, MSb first. As with three-byte instructions, the lower byte of a data word is presented first, followed by the upper byte.

As long as the $\overline{CS}$ pin is held low, the instruction continues to execute, automatically incrementing to the next register address in the SFR space and writing data from SI to, or outputting data on SO from, subsequent registers. When the end of a bank is reached, the address continues to the top of the next bank. Addresses continue to increment through the banks into the unbanked SFR area (addresses 80h through 9Fh), then wrap around to the start of Bank 0 (00h). The SFR bank value used by the BxSEL and RBSEL opcodes is not affected by the execution of unbanked SFR instructions.

There are four unbanked SFR opcodes, summarized in Table 4-5. Except for addressing, the unbanked SFR instructions are analogous to the banked SFR instructions. However, there are certain differences in their behavior with certain pointer registers, as noted below.

FIGURE 4-6: N-BYTE SPI OPCODE (UNBANKED SFR OPERATIONS)



4.6.2.1 WCRU Opcode

The Write Control Register Unbanked (WCRU) opcode starts with the opcode, '00100010' (22h), followed by the unbanked SFR register address during SPI clocks, 9 through 16. For example, to write to ECON2L at address 6Eh, the instruction would be '22h 6Eh', followed by the data to be written.

When the host controller is finished writing data, it should raise the $\overline{CS}$ line, putting the device in an inactive state and preparing it for the next SPI instruction. When finishing a WCRU transaction, ensure that adequate $\overline{CS}$ hold time is provided for the last write to complete before raising $\overline{CS}$.

Generally, WCRU can be executed on most register addresses, in any sequence and for any length. An important exception is when WCRU is used on any MAC or MII register. These registers must be written as whole 16-bit registers, low byte first (e.g., MACON1 must be written by first writing to MACON1L, then MACON1H). Writing only to the upper byte of a MAC or MII register results in a successful write to the upper register, while the lower register is written with indeterminate data. If a WCRU instruction is aborted by raising $\overline{CS}$ while writing to the upper byte of a MAC or MII register, neither the upper nor lower byte will be updated.

In addition, WCRU cannot be used to write to the SRAM buffer virtual data windows (EGPDATA, ERXDATA and EUDADATA). Writing to the buffer address indicated by the corresponding address pointers' attempts has no effect on the memory location, and the pointers do not auto-increment. To write to the SRAM buffer using the virtual data windows, always use the SRAM buffer opcodes (WGPDATA, WRXDATA and WUDADATA) instead.

4.6.2.2 RCRU Opcode

The Read Control Register Unbanked (RCRU) opcode starts with the opcode, '00100000' (20h), followed by the unbanked SFR register address during SPI clocks, 9 through 16. Continuing the previous example, to read ECON2L at address 6Eh, the complete two-byte instruction would be '20h 6Eh'.

Read operations can be performed on most register addresses, in any sequence and for any length. However, due to volatile register shadowing, it is recommended that the ERXHEADH:ERXHEADL register pair be read in sequence (low byte first) to obtain the correct value. See **Section 9.2 "Receiving Packets"** for additional information.

Similar to WCRU, RCRU cannot be used to read data from the SRAM buffer using the virtual data windows. Reading the buffer address indicated by the corresponding address pointers returns indeterminant data and the pointers do not auto-increment. To read from the buffer using the virtual data windows, always use the SRAM buffer opcodes (RGPDATA, RRXDATA and RUDADATA) instead.

4.6.2.3 BFSU and BFCU Opcodes

The Bit Field Set Unbanked (BFSU) and Bit Filed Clear Unbanked (BFCU) opcodes start with the opcode, '00100100' (24h) for BFSU, or '00100110' (26h) for BFCU, followed by the unbanked SFR register address during SPI clocks, 9 through 16. In terms of timing and automatic address increment, they behave almost identically to the WCRU opcode.

BFSU and BFCU function in the same manner as BFS and BFC, by setting or clearing individual bits in the target register through the use of a bit mask. They are also used in the same situations as BFS and BFC; namely, when it is necessary to manipulate a single control bit or interrupt flag in a dynamic situation, while avoiding the disruption of other bits. See **Section 4.6.1.3 "BFS and BFC Opcodes"** for a detailed explanation.

Note 1:	Unlike WCRU, BFSU and BFCU cannot be used to modify MAC or MII registers. Never use these opcodes on MAC and MII registers.
2:	BFSU and BFCU opcodes have no effect on any SFR in the unbanked region (addresses 80h through 9Fh).

TABLE 4-5: N-BYTE UNBANKED SFR INSTRUCTIONS

Instruction	Mnemonic	Opcode	Argument		
		1st Byte	2nd Byte	3rd Byte	Nth Byte
Read Control Register(s), Unbanked	RCRU	0010 0000	AAAA AAAA	xxxx xxxx	XXXX XXXX
Write Control Register(s), Unbanked	WCRU	0010 0010	AAAA AAAA	dddd dddd	DDDD DDDD
Bit Field(s) Set, Unbanked	BFSU	0010 0100	AAAA AAAA	dddd dddd	DDDD DDDD
Bit Field(s) Clear, Unbanked	BFCU	0010 0110	AAAA AAAA	dddd dddd	DDDD DDDD

Legend: x/X = read data (LSB/MSB), d/D = write data (LSB/MSB), A = unbanked SFR address. 'X' and 'D' are optional.

4.6.3 SRAM BUFFER OPERATIONS

The six N-byte SRAM instructions function in a similar manner to the banked SFR instructions, in that they use a single byte opcode to define the operation and target register. In terms of timing, they are virtually identical, as shown in Figure 4-7.

Like all other opcodes, data is presented on the SI pin, MSb first. For all instructions, the first byte of data is the opcode. If the instruction is a write opcode, the next bytes are the data to be written. For read instructions, the next bytes are don't cares.

For write instructions, the SO pin is actively driven with indeterminate '1's or '0's while the $\overline{CS}$ pin is driven low. For read instructions, random data is clocked out on SO during SCK clocks, 1 through 8. Starting with the 9th clock, data is clocked out byte-wise on SO, MSb first.

As long as the $\overline{CS}$ pin is held low, the instruction continues to execute, automatically incrementing to the next SRAM address according to the pointer wrapping rules described in **Section 3.5.5 "Indirect SRAM Buffer Access"**. The associated read or write pointer SFRs are automatically updated for each 8 SCK clocks. To terminate the read or write operation, the $\overline{CS}$ signal must be returned high.

There are 6 instructions divided equally between read and write instructions. They are summarized in Table 4-6.

FIGURE 4-7: N-BYTE SPI OPCODE (SRAM BUFFER OPERATIONS)

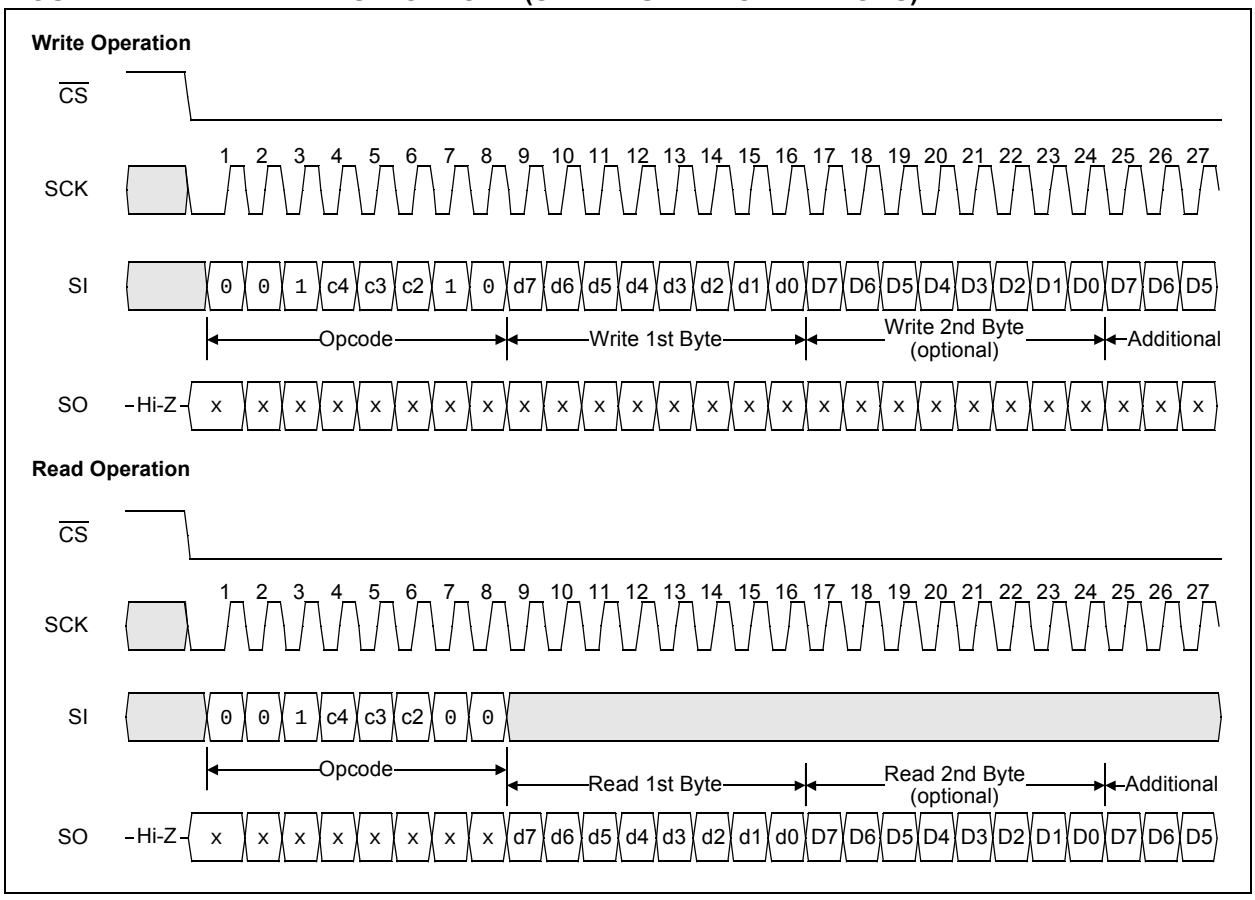


TABLE 4-6: N-BYTE SRAM INSTRUCTIONS

Instruction	Mnemonic	Opcode	Argument		
		1st Byte	2nd Byte	3rd Byte	Nth Byte
Read Data from EGPDATA	RGPDATA	0010 1000	xxxx xxxx	XXXX XXXX	XXXX XXXX
Write Data from EGPDATA	WGPDATA	0010 1010	dddd dddd	DDDD DDDD	DDDD DDDD
Read Data from ERXDATA	RRXDATA	0010 1100	xxxx xxxx	XXXX XXXX	XXXX XXXX
Write Data from ERXDATA	WRXDATA	0010 1110	dddd dddd	DDDD DDDD	DDDD DDDD
Read Data from EUDADATA	RUDADATA	0011 0000	xxxx xxxx	XXXX XXXX	XXXX XXXX
Write Data from EUDADATA	WUDADATA	0011 0010	dddd dddd	DDDD DDDD	DDDD DDDD

Legend: x/X = read data (LSB/MSB), d/D = write data (LSB/MSB). 'X' and 'D' are optional.

5.0 PARALLEL SLAVE PORT INTERFACE (PSP)

ENC424J600/624J600 devices are designed to interface directly with the parallel port available on many microcontrollers, including the Parallel Master Port (PMP) available on many Microchip PIC® microcontrollers. The Parallel Slave Port interface is highly flexible, and can communicate using either Intel® or Motorola® formats for read and write control strobes. In the event that a parallel port is not available on the host microcontroller, a software-managed parallel interface derived from general purpose I/O pins can be used.

When the PSP interface is enabled, the ENCX24J600 functions as a slave device on the parallel bus. The host controller must be configured to generate the destination or target address on the slave device, as well as the necessary port control signals.

5.1 Physical Implementation

The PSP interface is mutually exclusive with the serial interface. To enable the PSP and disable the SPI, tie the $\overline{INT}$/SPISEL pin to Vss through an external resistor.

The PSP interface can use from 12 to 34 pins, depending on the device pin count and the PSP operating mode. There are up to eight modes, covering the permutations of data widths, data/address multiplexing and bus strobe formats. The modes are selected by tieing each of the PSPCFG<4:0> pins to either VDD or VSS. The available combinations along with relative performance metrics are summarized in Table 5-1. Additional information on physical connections are provided in **Section 2.7.2 "PSP"**.

In PSP mode, the $\overline{CS}$/CS pin becomes the active-high Chip Select (CS) pin. A weak internal pull-down is automatically connected to the pin when the PSP interface is selected, preventing the pin from floating to an indeterminate state when an external Chip Select signal is absent.

When CS is in the inactive (logic-low) state, the AD15 (64-pin devices only) and AD<14:0> pins are placed in a high-impedance state and are 5V tolerant. This allows the ENCX24J600 to share a single parallel bus with other slave devices that function the same way while deselected. All other PSP pins, including the A<14:0> pins (64-pin devices only) and the port control strobes, are 5V tolerant inputs at all times. Inputs on these pins are ignored while the chip select pin is at logic low.

Unlike the SPI port, the use of chip select is optional with the PSP. The CS pin can be tied permanently to VDD if the parallel bus is not shared with other slave devices. This saves one I/O pin from the host controller while leaving the ENCX24J600 in a perpetually selected state.

TABLE 5-1: OPERATING MODES SUPPORTED BY THE PSP INTERFACE

PSP Mode	Availability		# Pins(1)		Data Width	Address/Data Multiplexing	Control Lines	Theoretical Performance @ 10 MHz (Mbit/s)
	44-pin	64-pin	Min	Max				
1		X	19	26	8 bit	No	CS, RD, WR	80
2		X	19	26	8 bit	No	CS, EN, R/$\overline{W}$	80
3		X	26	34	16 bit	No	CS, RD, WRL, WRH	160
4		X	26	34	16 bit	No	CS, R/$\overline{W}$, B0SEL, B1SEL	160
5	X	X	12	19	8 bit	Yes	AL, CS, RD, WR	<80
6	X	X	12	19	8 bit	Yes	AL, CS, EN, R/$\overline{W}$	<80
9		X	19	21	16 bit	Yes	AL, CS, RD, WRL, WRH	<80
10		X	19	21	16 bit	Yes	AL, CS, R/$\overline{W}$, B0SEL, B1SEL	<80

Note 1: Includes only address, data and port control strobes. $\overline{INT}$/SPISEL and PSPCFG pins used for mode configuration are not included.

5.2 Using the PSP Interface

Unlike the serial interface, the PSP interface does not use opcodes or a command architecture to control the device. Instead, the memory space is accessed directly using the addressing schemes described in **Section 3.1.2 "PSP Interface Maps"**. Control SFRs are read and written to directly, or manipulated through their accompanying Bit Set and Bit Clear registers.

In 16-bit modes, each address (from 0 to 16,384) points to a different word. The individual write high and write low strobes allow the upper or lower byte of each word to be written individually.

5.2.1 DIRECT AND INDIRECT SRAM BUFFER ACCESS

Direct addressing allows the host controller to access all SFRs and SRAM buffer addresses in the ENCX24J600 memory space directly. This provides the greatest flexibility and speed for accessing the SRAM buffer. However, this configuration requires up to 15 address pins to be driven by the host controller. This may be prohibitive in smaller, pin-constrained applications.

In Modes 1 through 6, it is possible to conserve six address pins by tying them to VDD. In this configuration, only the addresses corresponding to the SFR area of the memory space can be directly addressed. The SRAM buffer memory can still be accessed, but only through the EGPDATA, ERXDATA and EUDADATA data windows in the SFR space, described in **Section 3.5.5 "Indirect SRAM Buffer Access"**.

Indirect buffer access works well for Multiplexed modes, such as PSP Modes 5, 6, 9 and 10. In these modes, the auto-incrementing feature of the Data Window Pointers allows access to the buffer at speeds similar to byte-wise demultiplexed access, since a separate address phase in not required for each byte.

The 8-Bit PSP modes have separate addresses for the low and high bytes of each register. Since these modes, therefore, have a "longer" memory space (i.e., more individual addresses), indirect access requires 9 lines to address all registers between 7E00h and 7FFFh. In contrast, the 16-bit modes require only 8 lines to address all of the registers in their SFR range (3F00h to 3FFFh). Even so, using indirect access still saves six pins in either data width: AD<14:9> in 8-bit modes and AD<13:8> in 16-bit modes.

5.2.2 ADDRESS LATCHING

In Multiplexed Address/Data modes (PSP Modes 5 through 10), the ENCX24J600 implements an internal address latch. This allows a reduction in the total number of interface pins by multiplexing the data and addresses that need to be communicated onto a single bus.

In 8-bit modes, the address latch is implemented on all of the AD pins. In 16-bit modes, the address latch is implemented for only the AD<13:0> pins. Because it spans all required address lines, it is necessary to present the desired address to the ENCX24J600 for only a brief period while strobing the Address Latch (AL) pin. On 8-bit interfaces, where AD<14:8> are used exclusively for addressing, it is not necessary to drive these upper address lines with a valid address continually through read and write operations.

During operation, strobing the AL pin high and then low causes the address presented on the AD pins to be saved to the address latch. The address is retained for all future read and write operations. It is retained until either a POR event occurs or a subsequent write to the address latch occurs by restrobing AL. This allows multiple read and write requests to take place to the same address, without requiring multiple address latching operations.

The address latch does not auto-increment after accesses. However, by using the indirect buffer access method, it is possible to sequentially read or write an entire array of sequential SRAM locations without updating the address latch.

5.2.3 WRITE SELECT PINS

The 16-Bit PSP modes make use of either two write pins (WRL and WRH), or a R/$\overline{W}$ select and two Byte Lane (B0SEL and B1SEL) controls. When writing to the device, these pins allow the host controller to instruct whether to write only the low byte, only the high byte or both bytes.

If only one write select pin is available on the host controller, the high and low selection pins may be tied together to create a single 16-bit write strobe. When this is done, only word writes are possible. However, the host controller can still write single bytes when accessing the SRAM buffer through the EGPDATA, ERXDATA or EUDADATA Window registers, which always perform 8-bit accesses.

5.2.4 UNUSED INTERFACE PINS

Any unused PSP pins are placed in a high-impedance state, regardless of the state of the CS pin. For maximum ESD performance, it is recommended that unused interface pins not be allowed to float. Instead, it is recommended that unused interface pins be tied to either VSS or VDD.

5.2.5 PERFORMANCE CONSIDERATIONS

When using a 16-bit data bus width, all registers and direct access to SRAM can be accomplished through 16-bit accesses. Therefore, these modes are potentially twice as fast as their 8-bit equivalent parallel mode. However, accesses through the hardware-managed SRAM read/write registers, EGPDATA, ERXDATA and EUDADATA, are always 8-bit regardless of the interface used. Therefore, in many applications, it will not be practically feasible to transfer 16 bits of meaningful data for all bus transfer cycles.

When reading from the EGPDATA, ERXDATA and EUDADATA registers on an interface with a multiplexed address bus, it is possible to latch the address only once and then perform back-to-back reads or writes without performing additional address latch cycles. This can provide a significant performance improvement when sequentially reading or writing an array of data to/from the RAM. Due to this benefit, 8-Bit Multiplexed modes (Modes 5 and 6) approach the theoretical performance of the Demultiplexed PSP Modes 1 and 2.

5.3 PSP Modes

The eight PSP modes are selected using the PSPCFG pins. The address/data bus and port control connections differ between the modes, sometimes significantly, as do the timing relationships between address/data and control signals. Each of the modes is described in detail in the following sections.

5.3.1 MODE 1

PSP Mode 1 is an 8-bit, fully demultiplexed mode that is available on 64-pin devices only. The parallel interface consists of 8 bi-directional data pins (AD<7:0>) and 9 to 15 separate address pins (A<14:0>). To select PSP Mode 1, tie PSPCFG2, PSPCFG3 and PSPCFG4 to VSS. Figure 5-1 shows the connections required.

This mode uses active-high Read and Write strobes (RD and WR) in conjunction with a Chip Select (CS) signal. These three pins allow the host to select the device, then signal when a read operation is desired or when valid data is being presented to be written. The AD<7:0> pins stay in a high-impedance state any time CS or RD is low.

To perform a read operation:

1. Raise the CS line (if connected to the host).
2. Present the address to be read onto the address bus.
3. Raise the RD strobe and wait the required time for the access to occur.

When RD is raised high, the data bus begins to drive out indeterminate data for a brief period, then switches to the correct read data after the appropriate read access time has elapsed. When the RD strobe is lowered, AD<7:0> return to a high-impedance state.

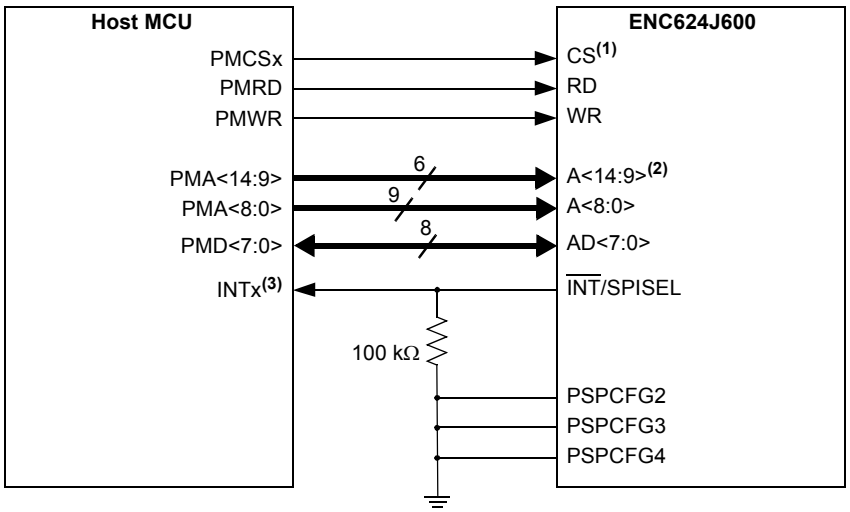
To perform a write operation:

1. Raise the CS line (if connected to the host).
2. Present the address onto the address bus.
3. Present the data on the data bus.
4. Strobe the WR signal high and then low.

For proper operation, do not raise RD and WR simultaneously while the ENCX24J600 is selected.

Sample timing diagrams for reading and writing data in this mode are provided in Figure 5-2 and Figure 5-3, respectively.

FIGURE 5-1: DEVICE CONNECTIONS FOR PSP MODE 1



Note 1: Use of the CS strobe from the controller is optional. If not used, tie CS to VDD.

2: Connect these pins when direct addressing of the entire SRAM buffer is required. Tie to VDD when only indirect addressing is desired.

3: Use of the external interrupt signal to the controller is optional.

FIGURE 5-2: MODE 1 READ OPERATION TIMING (TWO BYTES)

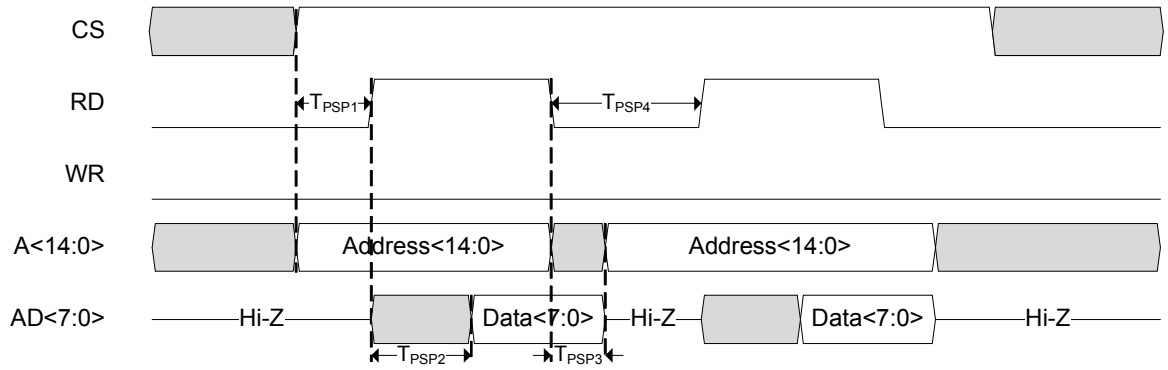
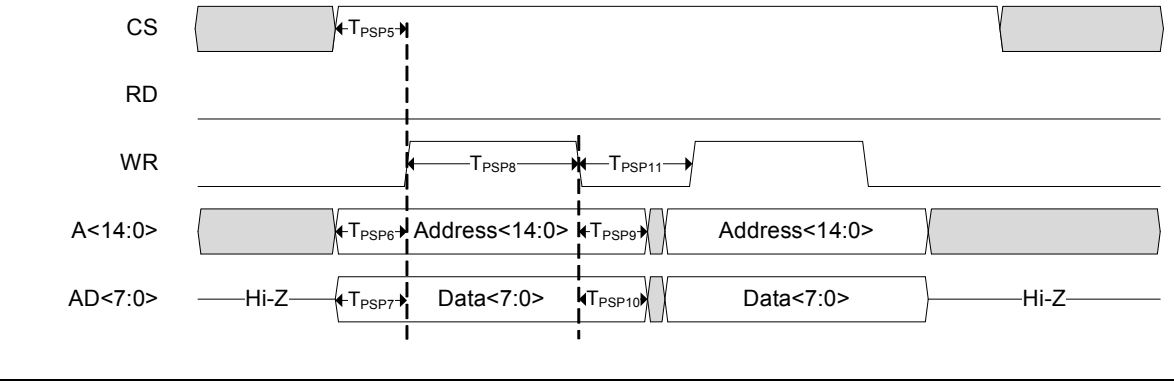


FIGURE 5-3: MODE 1 WRITE OPERATION TIMING (TWO BYTES)



5.3.2 MODE 2

PSP Mode 2 is also an 8-bit, fully demultiplexed mode that is available on 64-pin devices only. The parallel interface consists of 8 bidirectional data pins (AD<7:0>) and 9 to 15 separate address pins (A<14:0>). To select PSP Mode 2, tie PSPCFG2 and PSPCFG3 to VSS, while connecting PSPCFG4 to VDD. Figure 5-4 demonstrates connections required to use Mode 2.

This mode uses a combined Read/$\overline{\text{Write}}$ (R/$\overline{\text{W}}$) select, an Enable (EN) strobe pin and a separate Chip Select pin (CS). These three pins allow the host to select the device, indicate whether a read or write operation is desired and signal when valid data is being presented

A logic high signal on the R/$\overline{\text{W}}$ pin indicates that a read operation is to be performed when the EN strobe is asserted, while a logic low indicates that a write operation is to be performed. The state of R/$\overline{\text{W}}$ only affects the data bus state when the EN signal is active. When either CS, EN or R/$\overline{\text{W}}$ is driven low, the data bus stays in a high-impedance state.

To perform a read operation:

1. Raise the CS line (if connected to the host).
2. Raise the R/$\overline{\text{W}}$ signal.
3. Present the address to be read onto the address bus.
4. Raise the EN strobe.
5. Wait the required time for the access to occur.

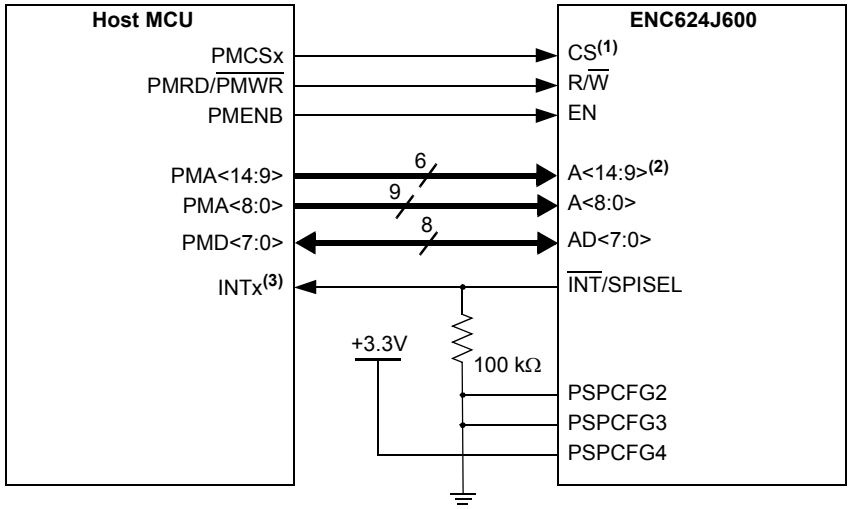
When EN is raised high, the data bus begins to drive out indeterminate data for a brief period, then switches to the correct read data after the appropriate read access time has elapsed. When the EN strobe is lowered, the data bus pins return to a high-impedance state.

To perform a write operation:

1. Raise the CS line (if connected to the host).
2. Lower the R/$\overline{\text{W}}$ signal.
3. Present the address onto the address bus.
4. Present the data on the data bus.
5. Strobe the EN signal high and then low.

Sample timing diagrams for reading and writing data in this mode are provided in Figure 5-5 and Figure 5-6, respectively.

FIGURE 5-4: DEVICE CONNECTIONS FOR PSP MODE 2



Note 1: Use of the CS strobe from the controller is optional. If not used, tie CS to VDD.

2: Connect these pins when direct addressing of the entire SRAM buffer is required. Tie to VDD when only indirect addressing is desired.

3: Use of the external interrupt signal to the controller is optional.

FIGURE 5-5: MODE 2 READ OPERATION TIMING (TWO BYTES)

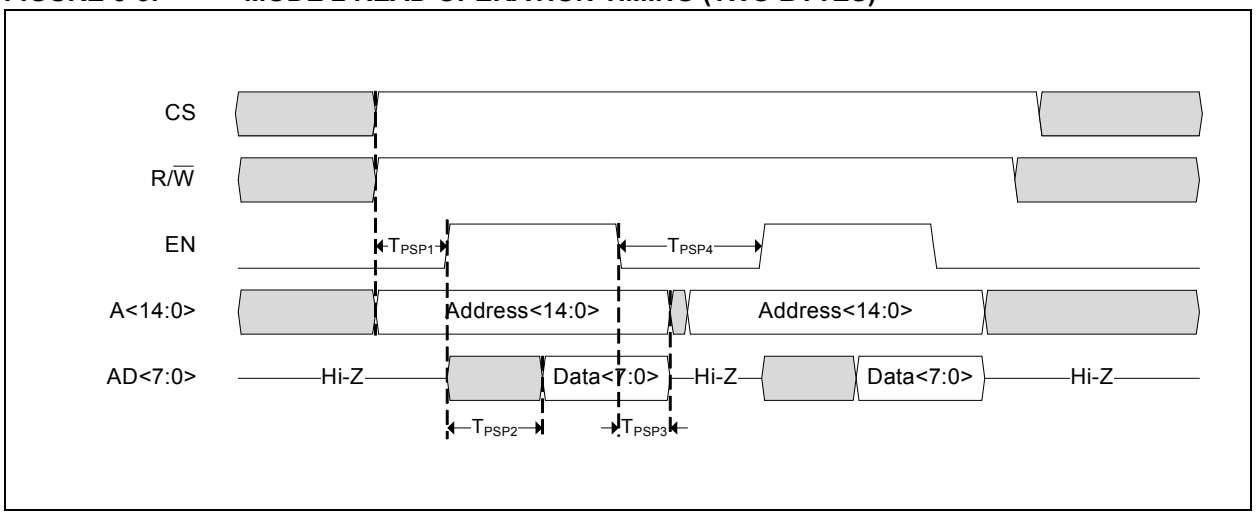
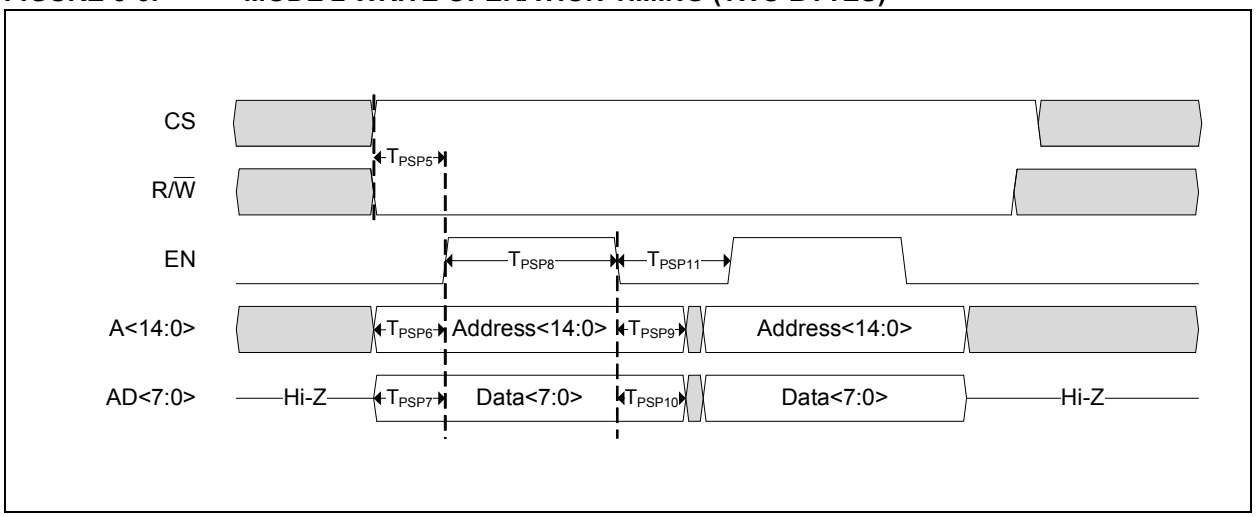


FIGURE 5-6: MODE 2 WRITE OPERATION TIMING (TWO BYTES)



5.3.3 MODE 3

PSP Mode 3 is a 16-bit, fully demultiplexed mode that is available on 64-pin devices only. The parallel interface consists of 16 bidirectional data pins (AD<15:0>) and 8 to 14 separate address pins (A<13:0>). To select PSP Mode 3, tie PSPCFG3 and PSPCFG4 to VSS, while connecting PSPCFG2 to VDD. Figure 5-7 shows the connections required.

An active-high RD strobe and two Write strobes (WRH and WRL) are utilized in conjunction with a separate Chip Select (CS). These four pins allow the host to select the device, then signal when a read operation is desired or when valid data is being presented to be written on either the low byte, high byte or both. For proper operation, do not assert CS and RD while simultaneously asserting either WRL or WRH.

In PSP Mode 3, AD<15:0> stay in a high-impedance state any time CS or RD are low.

To perform a read operation:

1. Raise the CS line (if connected to the host).
2. Present the address to be read onto the address bus.
3. Raise the RD strobe and wait the required time for the access to occur.

When RD is raised high, the data bus begins driving out indeterminate data for a brief period, then switches to the correct read data after the appropriate read access time has elapsed. When the RD strobe is lowered, the data pins will return to a high-impedance state.

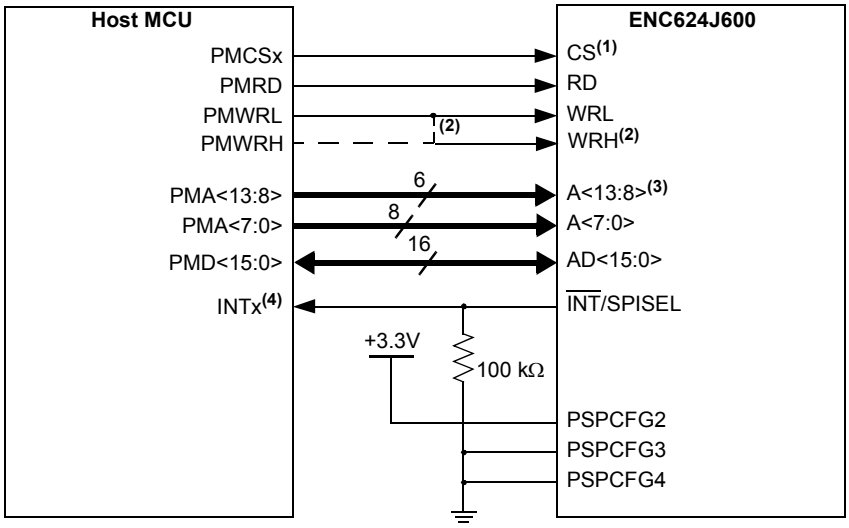
The device always outputs a full 16 bits of data for each read request. If only 8 bits of data are required, read the data from the correct pins (AD<15:8> or AD<7:0>) and discard the remaining byte.

To perform a write operation:

1. Raise the CS line (if connected to the host).
2. Present the address onto the A<13:0> address bus.
3. If writing to the low byte of the memory location, present the data on AD<7:0>, and strobe the WRL signal high and then low.
4. If writing to the high byte, present the data on the AD<15:8> and strobe the WRH signal.
5. If writing a whole word, strobe both WRL and WRH simultaneously.

Sample timing diagrams for reading and writing data in this mode are provided in Figure 5-8 and Figure 5-9, respectively.

FIGURE 5-7: DEVICE CONNECTIONS FOR PSP MODE 3



Note 1: Use of the CS strobe from the controller is optional. If not used, tie CS to VDD.

2: WRL and WRH may optionally be tied together to form a 16-bit write strobe. See **Section 5.2.3 "Write Select Pins"** for details.

3: Connect these pins when direct addressing of the entire SRAM buffer is required. Tie to VDD when only indirect addressing is desired.

4: Use of the external interrupt signal to the controller is optional.

FIGURE 5-8: MODE 3 READ OPERATION TIMING (FOUR BYTES)

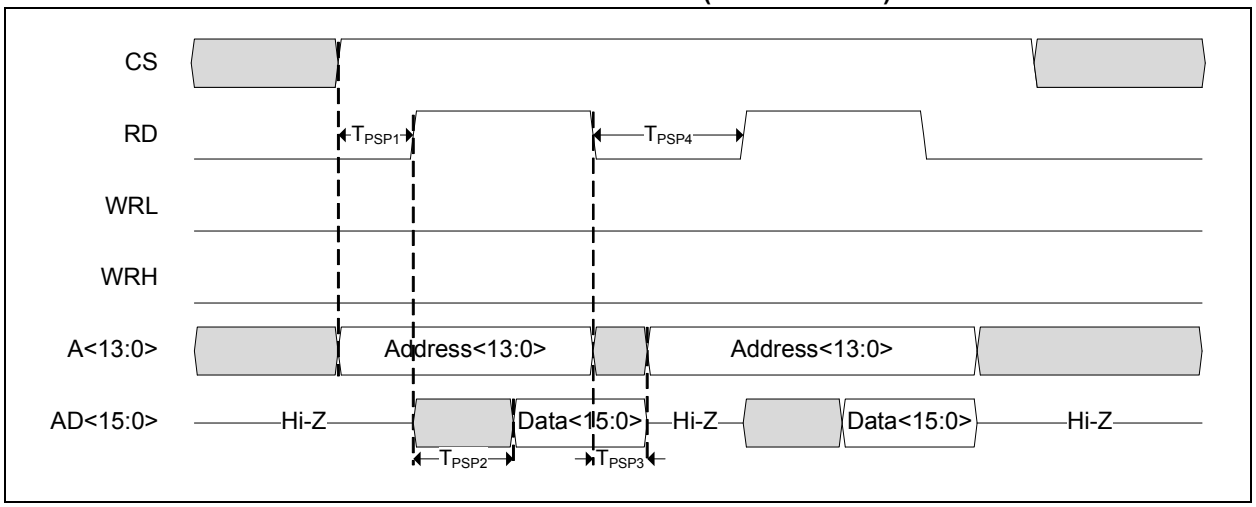
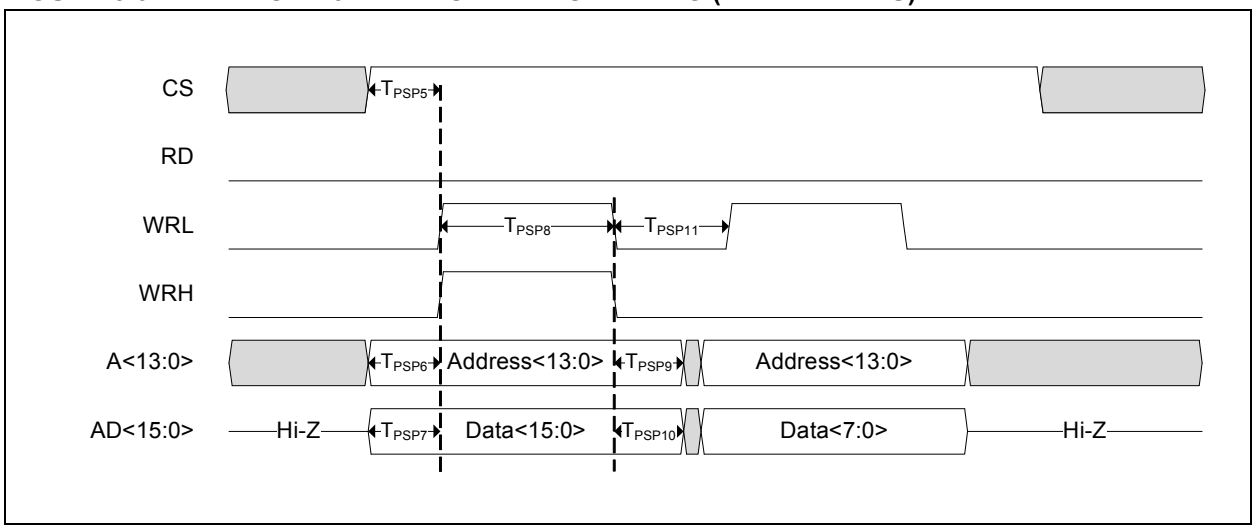


FIGURE 5-9: MODE 3 WRITE OPERATION TIMING (THREE BYTES)



5.3.4 MODE 4

PSP Mode 4 is also a 16-bit, fully demultiplexed mode that is available in 64-pin devices only. When using PSP Mode 4, the parallel interface consists of 16 bidirectional data pins (AD<15:0>) and 8 to 14 separate address pins (A<13:0>). To select PSP Mode 4, tie PSPCFG2 and PSPCFG4 to VDD, while connecting PSPCFG3 to VSS. Figure 5-10 shows the connections required.

This mode uses a combined Read/Write (R/$\overline{W}$) select, two Byte Select (B0SEL and B1SEL) lines and a separate Chip Select (CS) signal. These four pins allow the host to select the device, indicate whether a read or write operation is desired and signal when valid data is being presented for writing on either the low byte, high byte or both.

A logic-high signal on R/$\overline{W}$ indicates that a read operation is to be performed when either the B0SEL or B1SEL strobe is asserted, while a logic low signal indicates that a write operation is to be performed. The state of R/$\overline{W}$ only affects the data bus state when either B0SEL or B1SEL is active. When CS is driven low, R/$\overline{W}$ is driven low, or both B0SEL and B1SEL are driven low and the data bus stays in a high-impedance state.

To perform a read operation:

1. Raise the CS line (if connected to the host).
2. Raise the R/$\overline{W}$ signal.
3. Present the address to be read onto the address bus.
4. Raise one or both byte select strobes.

When either BxSEL pin is raised high, the data bus begins driving out indeterminate data for a brief period, then switches to the correct read data after the appropriate read access time has elapsed. When B0SEL and B1SEL are both low, the data bus pins return to a high-impedance state.

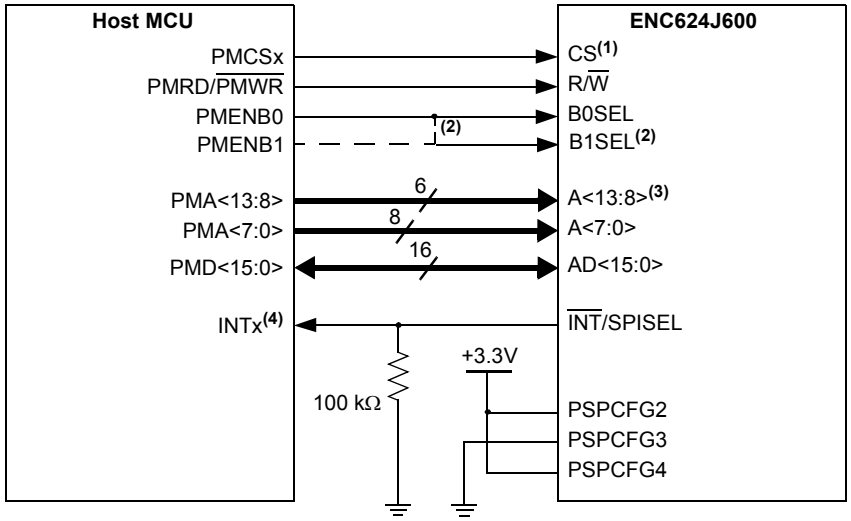
The device always outputs a full 16 bits of data for each read request, even if only one byte select is strobed. If only 8 bits of data are required, read the data from the correct pins (AD<15:8> or AD<7:0>) and discard the remaining byte.

To perform a write operation:

1. Raise the CS line (if connected to the host).
2. Lower R/$\overline{W}$.
3. Present the address onto the address bus.
4. If writing to the low byte of the memory location, present the data on the AD<7:0>; then strobe B0SEL high, then low.
5. If writing to the high byte, present the data on AD<15:8> and strobe B1SEL.
6. If writing a whole word, strobe both B0SEL and B1SEL simultaneously.

Sample timing diagrams for reading and writing data in this mode are provided in Figure 5-11 and Figure 5-12, respectively.

FIGURE 5-10: DEVICE CONNECTIONS FOR PSP MODE 4



Note 1: Use of the CS strobe from the controller is optional. If not used, tie CS to VDD.

2: B0SEL and B1SEL may optionally be tied together to form a 16-bit write strobe. See **Section 5.2.3 "Write Select Pins"** for details.

3: Connect these pins when direct addressing of the entire SRAM buffer is required. Tie to VDD when only indirect addressing is desired.

4: Use of the external interrupt signal to the controller is optional.

FIGURE 5-11: MODE 4 READ OPERATION TIMING (FOUR BYTES)

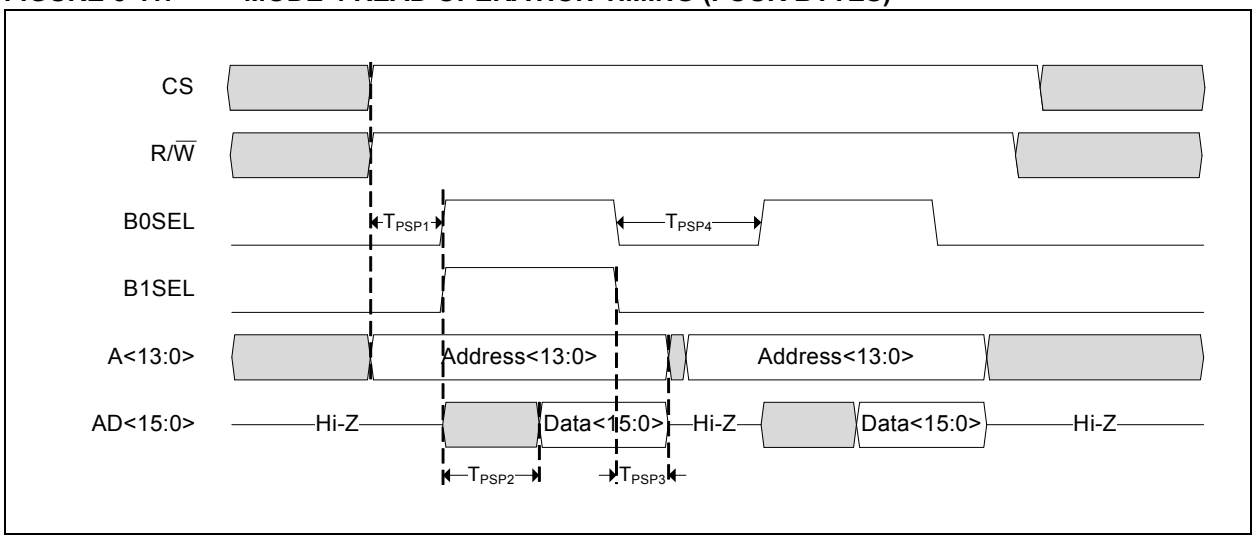
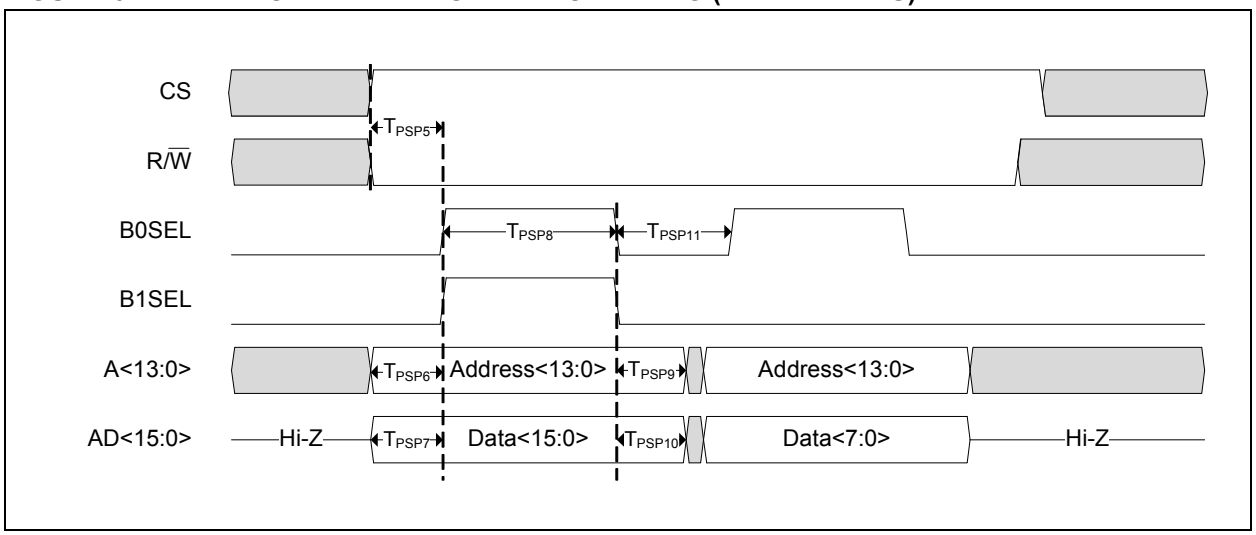


FIGURE 5-12: MODE 4 WRITE OPERATION TIMING (THREE BYTES)



5.3.5 MODE 5

PSP Mode 5 is an 8-bit, partially multiplexed mode that is available on all devices. The parallel interface consists of 8 multiplexed address and data pins (AD<7:0>), plus one required high address bit (AD8) and 6 optional address-only pins (AD<14:9>).

Selecting PSP Mode 5 differs between 44-pin and 64-pin devices, as shown in Figure 5-13. For the 44-pin ENC424J600, tie PSPCFG0 to VSS. For the 64-pin ENC624J600, tie PSPCFG1 and PSPCFG2 to VSS, and PSPCFG3 to VDD.

This mode uses active-high Read and Write (RD and WR) strobes, as well as separate Chip Select and Address Latch (CS and AL) lines. These four pins allow the host to select the device, latch an address, then indicate when a read or write operation is desired. For proper operation, treat the RD, WR and AL strobes as mutually exclusive whenever the ENCX24J600 is selected. Only raise one of these to logic high at any given time.

AD<14:8> are used as address inputs only, and are therefore, always left in a high-impedance state. When CS or RD is driven low, the multiplexed AD<7:0> pins stay in a high-impedance state.

To perform a read operation:

1. Raise CS (if connected to the host).
2. Present the address to read from on AD<14:0>.
3. Strobe the AL pin high and low.
4. Set the host controller's AD<7:0> bus pins as inputs.
5. Raise RD.

The AD<7:0> bus begins driving out indeterminate data for a brief period, then switches to the correct read data after the appropriate read access time has elapsed. When RD is lowered, the AD<7:0> pins return to a high-impedance state.

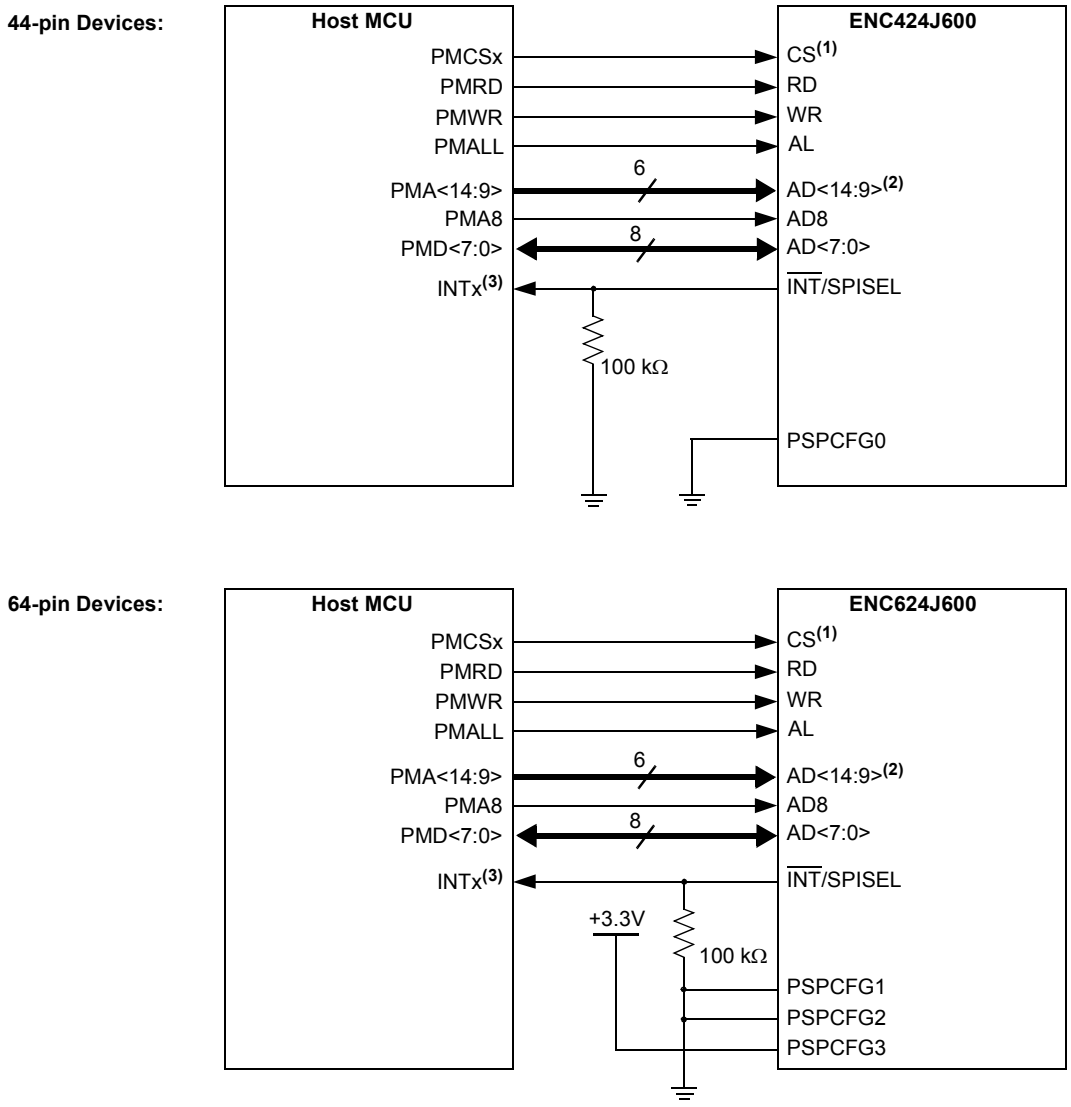
To perform a write operation:

1. Raise CS (if connected to the host).
2. Present the address to write to on AD<14:0>.
3. Strobe the AL pin.
4. Change the data on AD<7:0> from the lower address byte to the data to be written.
5. Strobe WR high and then low.

If a subsequent read or write of the same memory address is desired, it is possible to restrobe RD or WR without going through another address latch cycle.

Sample timing diagrams for reading and writing data in this mode are provided in Figure 5-14 and Figure 5-15, respectively.

FIGURE 5-13: DEVICE CONNECTIONS FOR PSP MODE 5

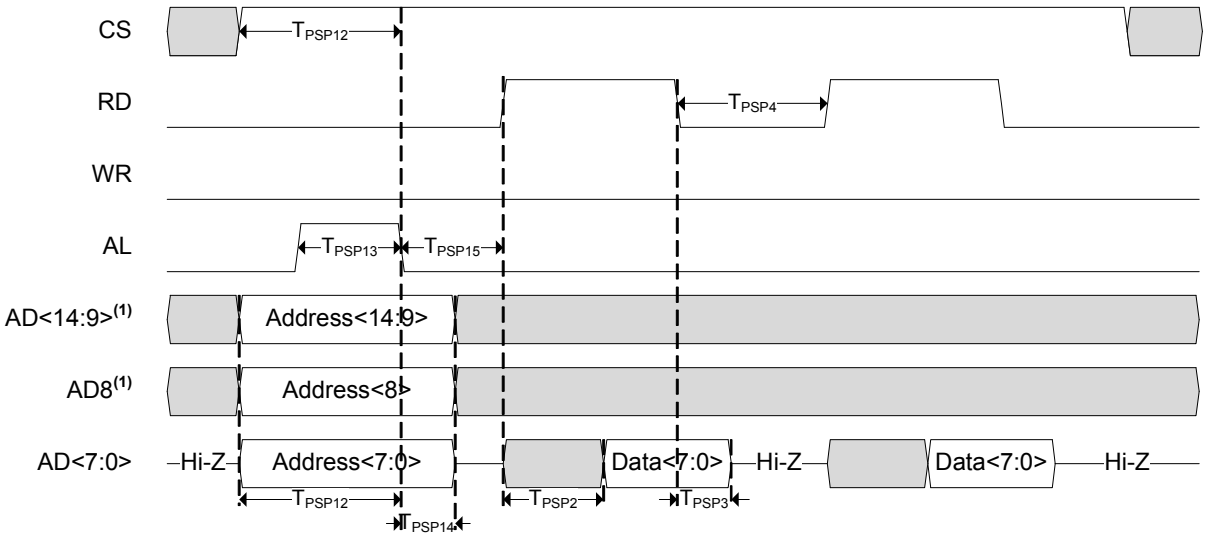


Note 1: Use of the CS strobe from the controller is optional. If not used, tie CS to VDD.

2: Connect these pins when direct addressing of the entire SRAM buffer is required. Tie to VDD when only indirect addressing is desired.

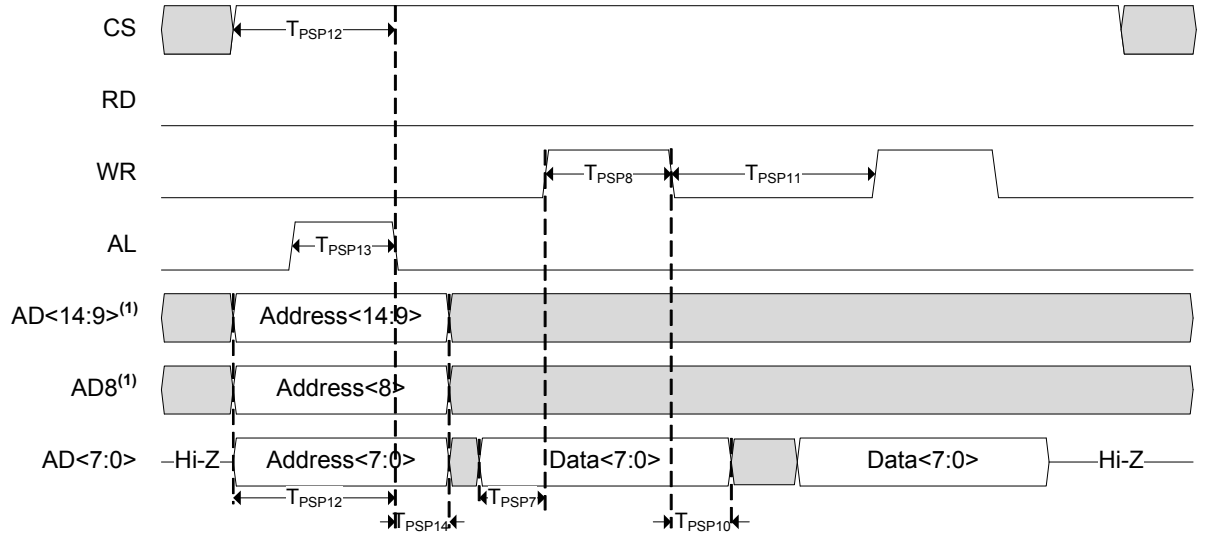
3: Use of the external interrupt signal to the controller is optional.

FIGURE 5-14: MODE 5 READ OPERATION TIMING (TWO BYTES – SAME ADDRESS)



Note 1: AD8 must be driven by the host controller. AD<14:9> may be tied to logic high when only indirect access is desired.

FIGURE 5-15: MODE 5 WRITE OPERATION TIMING (TWO BYTES – SAME ADDRESS)



Note 1: AD8 must be driven by the host controller. AD<14:9> may be tied to logic high when only indirect access is desired.

5.3.6 MODE 6

PSP Mode 6 is also an 8-bit, partially multiplexed mode that is available on all devices. The parallel interface consists of 8 multiplexed address and data pins (AD<7:0>), plus one required high address bit (AD8) and 6 optional address-only pins (AD<14:9>).

Selecting PSP Mode 6 differs between 44-pin and 64-pin devices, as shown in Figure 5-16. For the 44-pin ENC424J600, tie PSPCFG0 to VDD. For the 64-pin ENC624J600, tie PSPCFG1 and PSPCFG3 to VDD, and PSPCFG2 to VSS.

This mode uses a combined Read/Write (R/$\overline{W}$) select, an Enable (EN) strobe and separate Chip Select (CS) and Address Latch (AL) lines. These four pins allow the host to select the device, latch an address, select either a read or write operation, then assert the Enable pin when a read is requested or the data to be written is valid. For proper operation, do not assert EN and AL simultaneously while the ENCX24J600 is selected.

AD<14:8> are used as address inputs only, and are therefore, always left in a high-impedance state. When CS, R/$\overline{W}$ or EN is driven low, the multiplexed AD<7:0> pins stay in a high-impedance state.

To perform a read operation:

1. Raise CS (if connected to the host).
2. Present the address to read from on AD<14:0>.
3. Strobe AL high and then low.
4. Set the host controller's AD<7:0> bus pins as inputs.
5. Raise R/$\overline{W}$.
6. Raise the EN strobe.

The AD<7:0> bus begins driving out indeterminate data for a brief period, then switches to the correct read data after the appropriate read access time has elapsed. When EN is lowered, the multiplexed AD<7:0> pins return to a high-impedance state.

To perform a write operation:

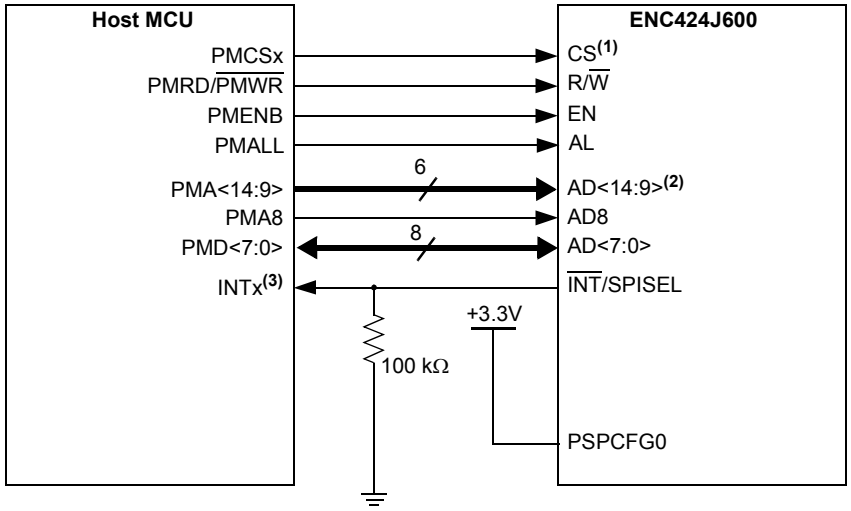
1. Raise CS (if connected to the host).
2. Present the address to write to on AD<14:0>.
3. Strobe AL.
4. Lower R/$\overline{W}$.
5. Change the data on AD<7:0> from the lower address byte to the data to be written.
6. Strobe EN high, then low.

If a subsequent read or write of the same memory address is desired, it is possible to restrobe EN without going through another address latch cycle.

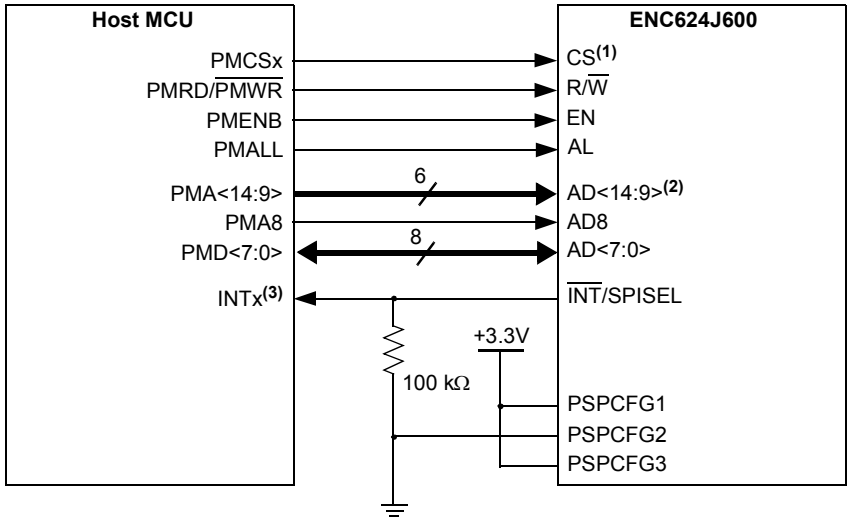
Sample timing diagrams for reading and writing data in this mode are provided in Figure 5-17 and Figure 5-18, respectively.

FIGURE 5-16: DEVICE CONNECTIONS FOR PSP MODE 6

44-pin Devices:



64-pin Devices:

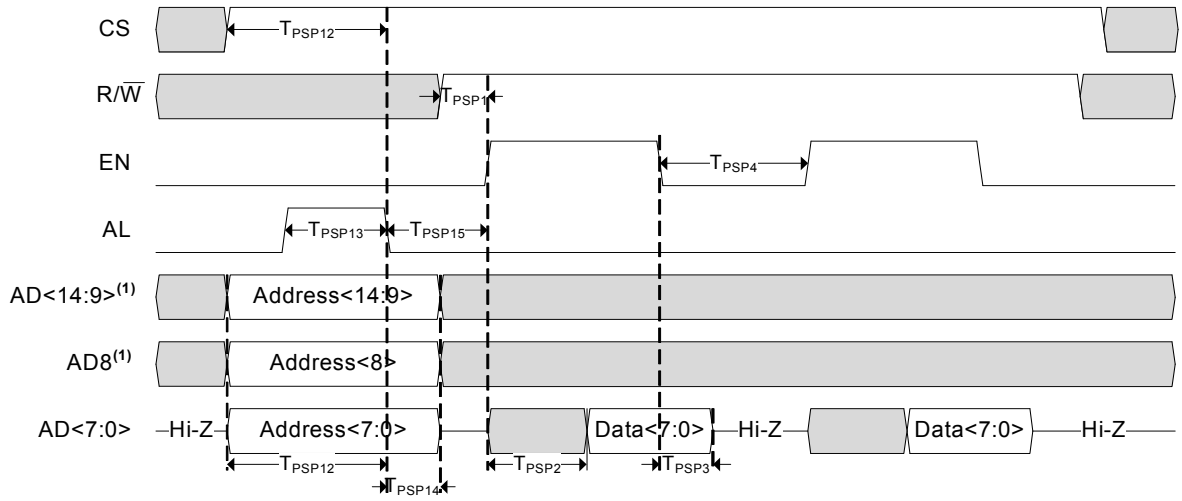


Note 1: Use of the CS strobe from the controller is optional. If not used, tie CS to VDD.

2: Connect these pins when direct addressing of the entire SRAM buffer is required. Tie to VDD when only indirect addressing is desired.

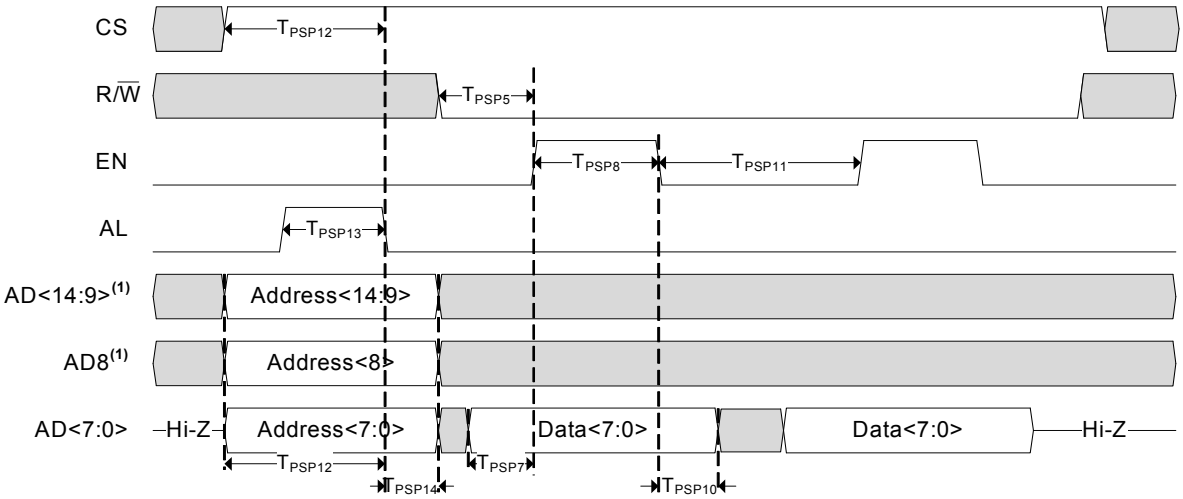
3: Use of the external interrupt signal to the controller is optional.

FIGURE 5-17: MODE 6 READ OPERATION TIMING (TWO BYTES – SAME ADDRESS)



Note 1: AD8 must be driven by the host controller. AD<14:9> may be tied to logic high when only indirect access is desired.

FIGURE 5-18: MODE 6 WRITE OPERATION TIMING (TWO BYTES – SAME ADDRESS)



Note 1: AD8 must be driven by the host controller. AD<14:9> may be tied to logic high when only indirect access is desired.

5.3.7 MODE 9

PSP Mode 9 is a 16-bit, fully-multiplexed mode that is available on 64-pin devices only. The parallel interface consists of 16 bidirectional data pins (AD<15:0>); the lower 14 (AD<13:0>) also function as address pins. To select PSP Mode 9, tie PSPCFG2 and PSPCFG3 to VDD, while connecting PSPCFG1 to VSS. Figure 5-19 shows the connections required.

This mode uses an active-high Read (RD) strobe and two Write (WRH and WRL) strobes in conjunction with separate Chip Select (CS) and Address Latch (AL) inputs. These five pins allow the host to select the device, latch an address and then signal when a read operation is desired or when valid data is being presented to be written to either the low byte, high byte or both. For proper operation while the ENCX24J600 is selected, do not assert RD or AL while simultaneously asserting either WRL or WRH.

AD<15:0> stay in a high-impedance state any time CS or RD is low.

To perform a read operation:

1. Raise CS (if connected to the host).
2. Present the address to read from on AD<13:0>.
3. Strobe AL high, then low.
4. Set the host controller's AD<15:0> bus pins as inputs.
5. Raise RD.

The AD<15:0> bus begins driving out indeterminate data for a brief period, then switches to the correct read data after the appropriate read access time has elapsed. When RD is lowered, the AD<15:0> pins return to a high-impedance state.

The device always outputs a full 16 bits of data for each read request. If only 8 bits of data are required, read the data from the correct pins (AD<15:8> or AD<7:0>) and discard the remaining byte.

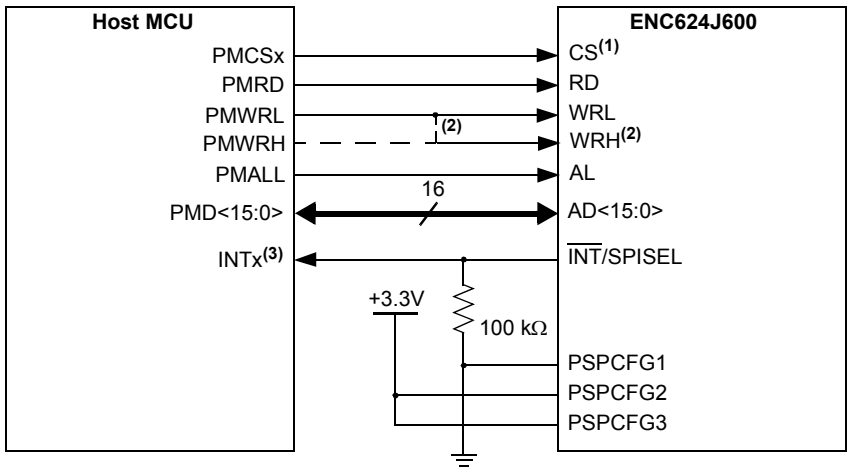
To perform a write operation:

1. Raise CS (if connected to the host).
2. Present the address to write to on AD<13:0>.
3. Strobe AL.
4. If writing to the low byte of the memory location, present the data on AD<7:0>, then strobe WRL high, then low.
5. If writing to the high byte, present the data on AD<15:8>, then strobe WRH.
6. If writing a whole word, strobe both WRL and WRH simultaneously.

If a subsequent read or write of the same memory address is desired, it is possible to restrobe RD, WRL or WRH without going through another address latch cycle.

Sample timing diagrams for reading and writing data in this mode are provided in Figure 5-20 and Figure 5-21, respectively.

FIGURE 5-19: DEVICE CONNECTIONS FOR PSP MODE 9



Note 1: Use of the CS strobe from the controller is optional. If not used, tie CS to VDD.

2: WRL and WRH may optionally be tied together to form a 16-bit write strobe. See **Section 5.2.3 "Write Select Pins"** for details.

3: Use of the external interrupt signal to the controller is optional.

FIGURE 5-20: MODE 9 READ OPERATION TIMING (FOUR BYTES – SAME ADDRESS)

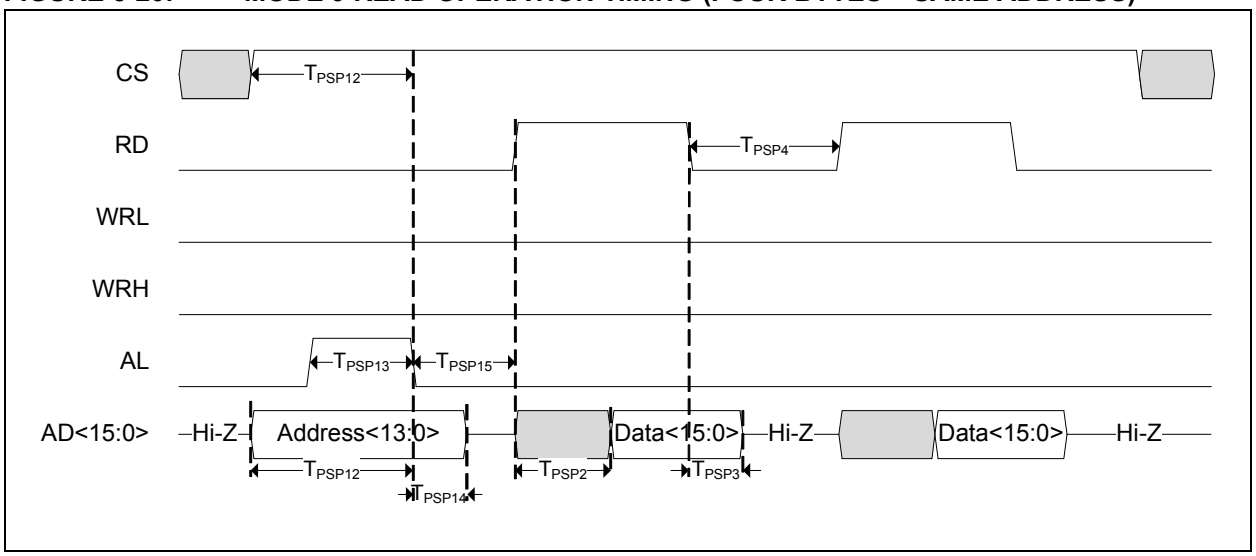
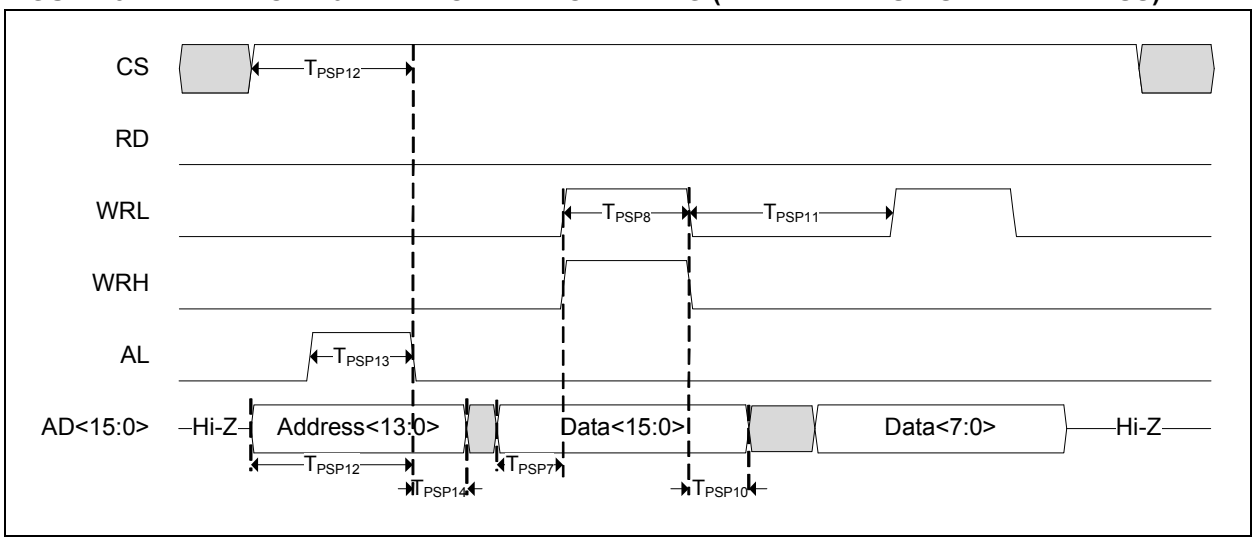


FIGURE 5-21: MODE 9 WRITE OPERATION TIMING (THREE BYTES – SAME ADDRESS)



5.3.8 MODE 10

PSP Mode 10 is also a 16-bit, fully-multiplexed mode that is available on 64-pin devices only. The parallel interface consists of 16 bidirectional data pins (AD<15:0>); the lower 14 (AD<13:0>) also function as address pins. To select PSP Mode 10, tie PSPCFG1, PSPCFG2 and PSPCFG3 to VDD. Figure 5-22 shows the connections required.

This mode uses an active-high Read/Write (R/$\overline{W}$) select and two Byte Select (B0SEL and B1SEL) strobes in conjunction with separate Chip Select (CS) and Address Latch (AL) inputs. These five pins allow the host to select the device, latch an address, select either a read or write operation, then assert the proper Byte Select strobe(s) to perform the operation.

A logic high signal on the R/$\overline{W}$ pin indicates that a read operation is to be performed when either the B0SEL or B1SEL strobe is asserted, while a logic low signal indicates that a write operation is to be performed. For proper operation while the ENCX24J600 is selected, the host controller should not assert AL while simultaneously asserting either B0SEL or B1SEL.

The state of R/$\overline{W}$ only affects the AD<15:0> bus state when either B0SEL or B1SEL is active. When CS is driven low, R/$\overline{W}$ is driven low, or both B0SEL and B1SEL are driven low, AD<15:0> stays in a high-impedance state.

To perform a read operation:

1. Raise CS (if connected to the host).
2. Present the address to be read onto AD<13:0>.
3. Strobe AL high, then low.
4. Raise R/$\overline{W}$.
5. Set the host controller's AD<15:0> bus pins as inputs.
6. Raise either B0SEL or B1SEL, or both.

When either BxSEL pin is raised high, the AD<15:0> bus begins driving out indeterminate data for a brief period, then switches to the correct read data after the appropriate read access time has elapsed. When B0SEL and B1SEL are both low, AD<15:0> return to a high-impedance state.

The device always outputs a full 16 bits of data for each read request, even if only one byte select is strobed. If only 8 bits of data are required, read the data from the correct pins (AD<15:8> or AD<7:0>) and discard the remaining byte.

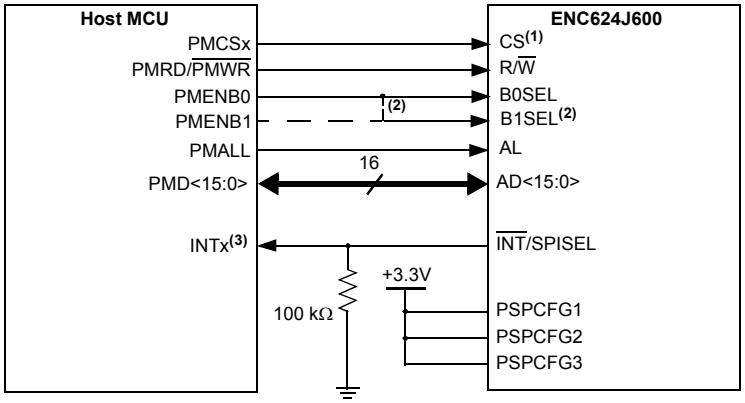
To perform a write operation:

1. Raise CS (if connected to the host).
2. Present the address to write to on AD<13:0>.
3. Strobe AL.
4. Lower R/$\overline{W}$.
5. If writing to the low byte of the memory location, present the data on AD<7:0>, then strobe B0SEL.
6. If writing to the high byte, present the data on AD<15:8>, then strobe the B1SEL signal.
7. If writing a whole word, strobe both B0SEL and B1SEL simultaneously.

If a subsequent read or write of the same memory address is desired, it is possible to restrobe B0SEL or B1SEL without going through another address latch cycle.

Sample timing diagrams for reading and writing data in this mode are provided in Figure 5-23 and Figure 5-24, respectively.

FIGURE 5-22: DEVICE CONNECTIONS FOR PSP MODE 1



Note 1: Use of the CS strobe from the controller is optional. If not used, tie CS to VDD.

2: B0SEL and B1SEL may optionally be tied together to form a 16-bit write strobe. See **Section 5.2.3 "Write Select Pins"** for details.

3: Use of the external interrupt signal to the controller is optional.

FIGURE 5-23: MODE 10 READ OPERATION TIMING (FOUR BYTES – SAME ADDRESS)

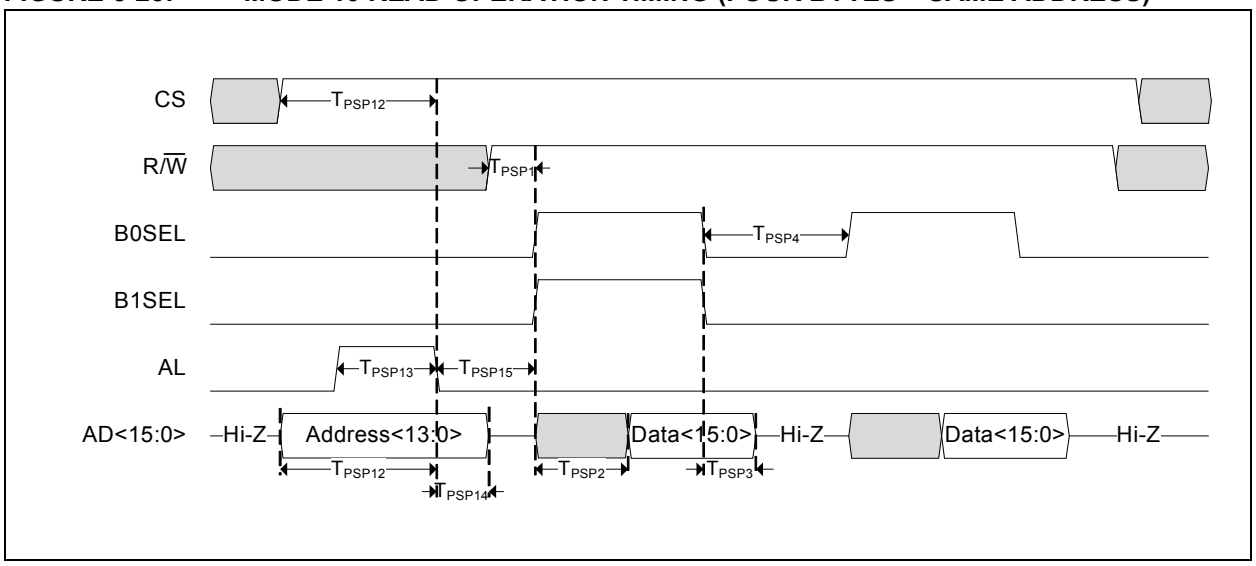
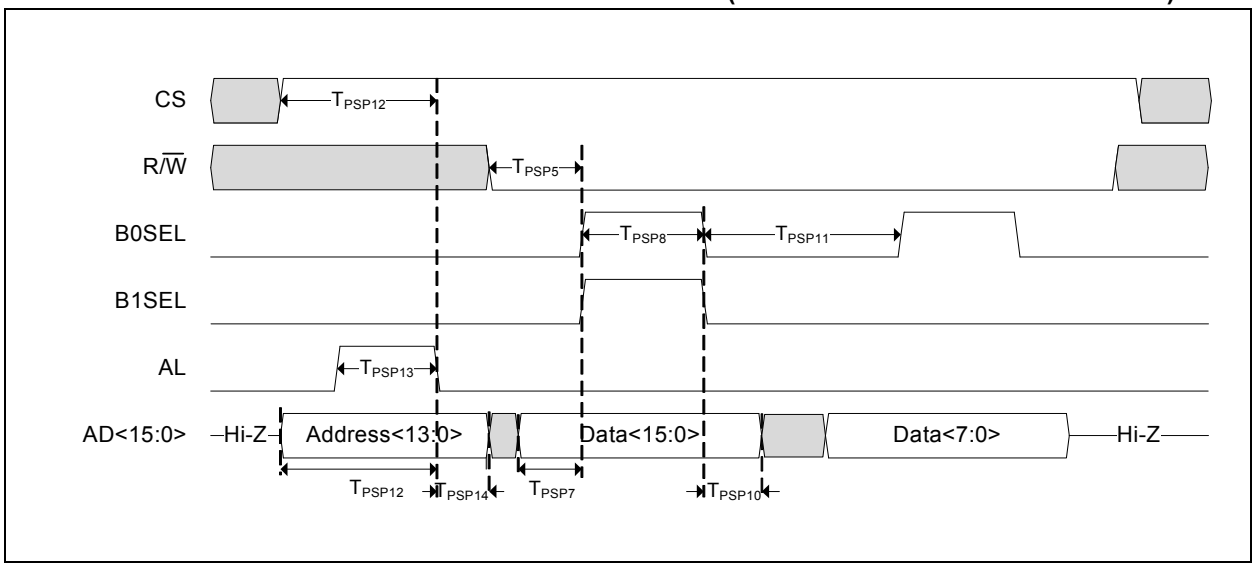


FIGURE 5-24: MODE 10 WRITE OPERATION TIMING (THREE BYTES – SAME ADDRESS)



6.0 ETHERNET OVERVIEW

Before discussing the use of ENC424J600/624J600 devices in Ethernet applications, it may be helpful to review the structure of a typical data frame. For more detailed information, refer to IEEE 802.3 Standard, which defines the Ethernet protocol, or to Microchip Application Note *AN1120,* "*Ethernet Theory of Operation*".

6.1 Frame Format

Ethernet communications utilize a series of frames to transmit data between nodes. (These frames are also commonly referred to as "packets", and in the context of this document, the two terms will be used interchangeably.) Compliant Ethernet frames are between 64 and 1518 bytes long. They consist of five or six different fields: a destination MAC address, source MAC address, type/length field, data payload, optional padding field and a Cyclic Redundancy Check (CRC). Additionally, when transmitted on the Ethernet medium, a start of stream/preamble field and a Start-Of-Frame (SOF) delimiter byte are appended to the beginning of the Ethernet frame. Thus, traffic seen on the twisted-pair cabling will appear as shown in Figure 6-1.

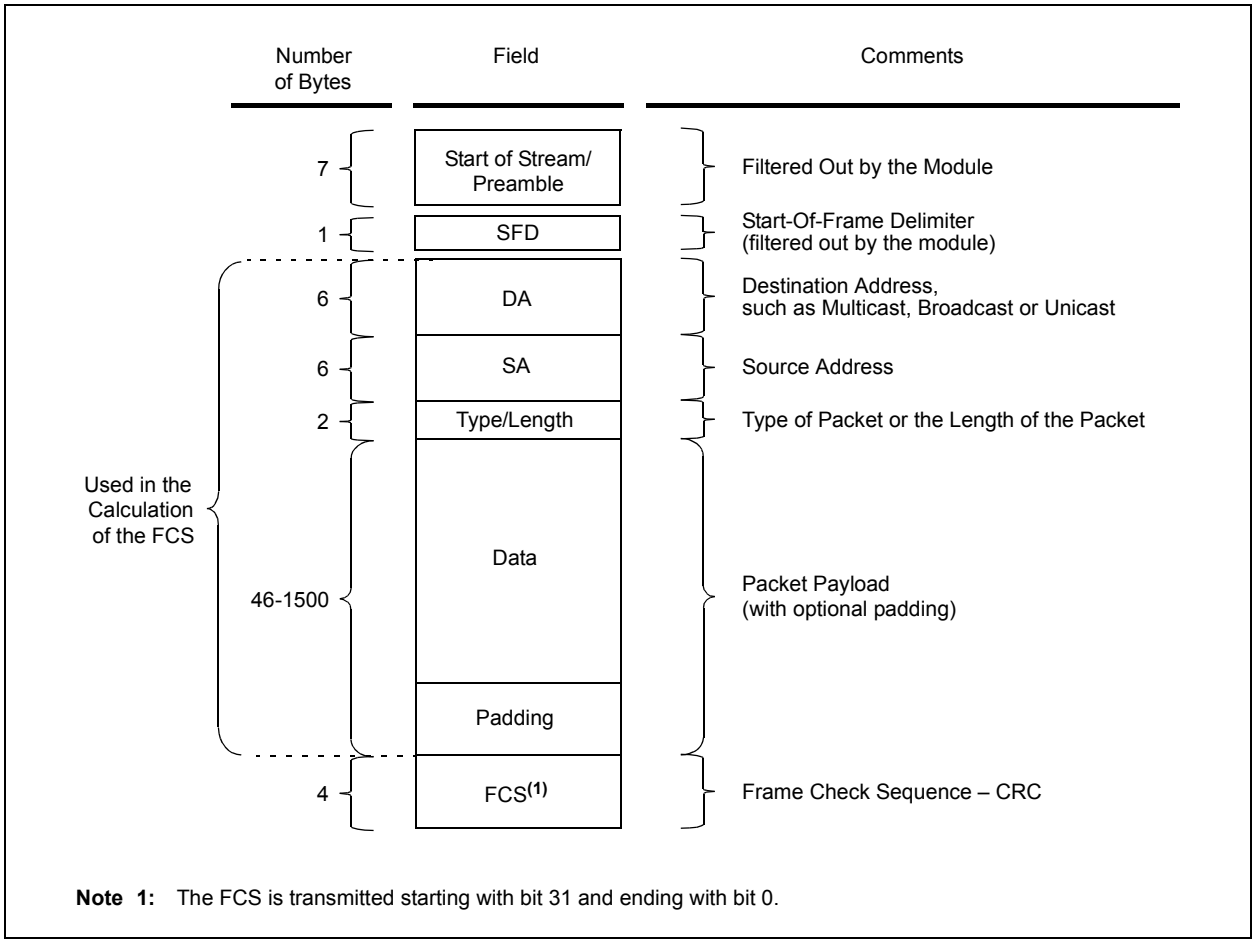
6.1.1 START OF STREAM/PREAMBLE AND START-OF-FRAME DELIMITER

When using ENC424J600/624J600 devices, the start of stream/preamble and Start-Of-Frame delimiter fields are automatically generated for transmitted frames and stripped from received ones. These bytes are not written to the data buffer and the host controller does not need to account for these bytes.

6.1.2 DESTINATION ADDRESS

The destination address is a 6-byte field containing the MAC address of the device to which the frame is directed. If the Least Significant bit in the first byte of this address is clear (i.e., the first byte of the address is even), the address is a Unicast address. For example, 00-00-BA-BE-F0-0D and 32-45-DE-AD-BE-EF Unicast addresses, while 01-00-BA-BE-F0-0D and 33-45-DE-AD-BE-EF are not. Frames with a Unicast destination are designated for usage by the addressed node only.

FIGURE 6-1: ETHERNET PACKET FORMAT



Note 1: The FCS is transmitted starting with bit 31 and ending with bit 0.

If the Least Significant bit in the first byte of this address is set (i.e., the byte is odd), the address is a Multicast destination. From the previous example, 01-00-BA-BE-F0-0D and 33-45-DE-AD-BE-EF are Multicast addresses. Multicast frames are designated for use by a selected group of Ethernet nodes. The Multicast address, FF-FF-FF-FF-FF-FF, is reserved; it is known as the Broadcast address and is directed to all nodes on the network.

ENC424J600/624J600 devices incorporate several packet filters which can be configured to accept or discard Unicast, Multicast and/or Broadcast frames. For details about these and other receive filters, refer to **Section 10.0 "Receive Filters"**. When transmitting frames, the host controller is responsible for writing the desired destination address into the transmit buffer.

6.1.3 SOURCE ADDRESS

The source address is a 6-byte field containing the MAC address of the node which transmitted the Ethernet frame. Every Ethernet device must have a globally unique MAC address. Each ENC424J600/624J600 device has a unique address which is loaded into the MAADR registers on power-up. This value can be used as is, or the registers may be reconfigured with a different address.

6.1.4 TYPE/LENGTH

The type/length field is a 2-byte field indicating the protocol to which the frame belongs. Applications using standards such as Internet Protocol (IP) or Address Resolution Protocol (ARP) should use the type code specified in the appropriate standards document. Alternately, this field can be used as a length field when implementing proprietary networks. Typically, any value of 1500 (05DCh) or smaller is considered to be a length field and specifies the amount of non-padding data which follows in the data field.

6.1.5 DATA

The data field typically consists of between 0 and 1500 bytes of payload data for each frame. ENC424J600/624J600 devices are capable of transmitting and receiving frames larger than this when the Huge Frame Enable bit, HFRMEN (MACON2<2>), is set. However, these larger data frames violate Ethernet specifications and will likely be dropped by most Ethernet nodes.

6.1.6 PADDING

The padding field is a variable length field appended to meet IEEE 802.3 specification requirements when transmitting small data payloads. As mentioned, the minimum Ethernet frame size is 64 bytes. Removing the 18 bytes of address and type information, and the terminating 4-byte CRC, leaves a minimum of 46 bytes. Smaller frames must be padded to fill this space.

When transmitting frames, ENC424J600/624J600 devices can automatically generate zero padding if the PADCFG<2:0> bits (MACON2<7:5>) are configured to do so. Otherwise, the application must append the appropriate padding. The device will not prevent the transmission of these "runt" frames if the host commands such an action, but the frame is likely to be dropped by other nodes.

When receiving frames, ENC424J600/624J600 devices accept and write all padding to the receive buffer. Frames shorter than the required 64 bytes can optionally be filtered by the Runt Error Reject filter, described in **Section 10.4 "Runt Error Rejection Filter"**.

6.1.7 CRC

The CRC is a 4-byte field containing a standard 32-bit CRC calculated over the destination, source, type, data and padding fields. It allows for the detection of transmission errors.

When transmitting frames, ENC424J600/624J600 devices can automatically generate and append a valid CRC if the PADCFG<2:0> bits are configured to do so. Otherwise, the host controller must generate and append this value. It is strongly recommended that the PADCFG bits be configured so that the hardware automatically manages this field.

When receiving frames, ENC424J600/624J600 devices accept and write the CRC field to the receive buffer. Frames with invalid CRC values can be discarded by the CRC Error Rejection filter, described in **Section 10.3 "CRC Error Rejection Filter"**.

7.0 RESET

ENC424J600/624J600 differentiates between five types of Resets:

- Power-on Reset (POR)
- System Reset
- Transmit Only Reset
- Receive Only Reset
- PHY Subsystem Reset

A simplified block diagram of the on-chip Reset circuit is shown in Figure 7-1.

7.1 Power-on Reset

Power-on Reset occurs when VDD rises above VPOR. This allows the device to start in the initialized state when VDD is adequate for the device's digital logic to operate correctly. The POR circuitry is always enabled.

To ensure proper POR operation, the application circuit must meet the specified minimum rise rate of VDD (SVDD, DC parameter D003).

After a Power-on Reset, the contents of the SRAM buffer and cryptographic memories are unknown. However, all registers will be loaded with their specified Reset values. The PHY and other logic should still not be accessed immediately after the POR. See **Section 8.1 "Reset"** for the recommended Reset procedure.

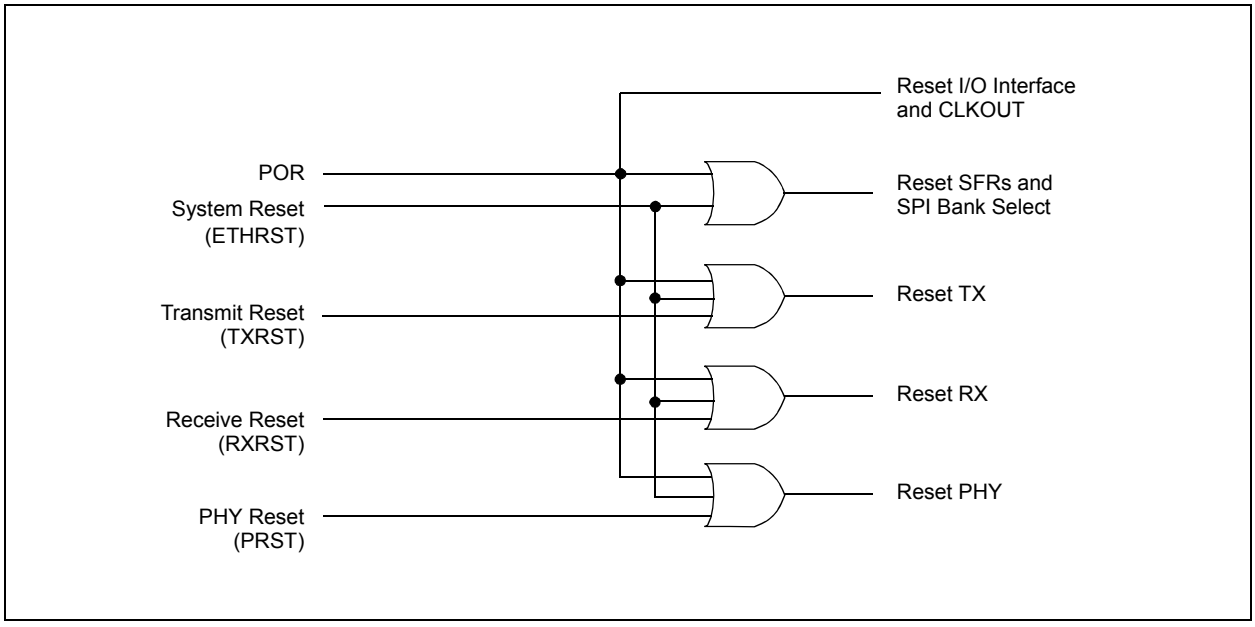
7.2 System Reset

A System Reset reverts all registers back to their default Reset values, with the exception of COCON<3:0> (ECON2<11:8>), which controls the frequency output on CLKOUT. All transmit, receive, MAC, PHY, DMA and cryptographic logic are reset. Additionally, if the SPI interface is used, the current internal bank selection is reset to Bank 0. The packet buffer, cryptographic memories and the PSP address latch used in Multiplexed Parallel modes are unaffected by a System Reset.

To initiate a System Reset, set the ETHRST bit (ECON2<4>). The bit is automatically cleared by hardware. After setting ETHRST, a delay of 25 μs is required before the ENCX24J600 can be accessed again through the SPI or PSP interfaces. Additionally, all PHY registers and status bits derived from the PHY should not be accessed or used for an additional period of 256 μs.

A System Reset does not cause the SPISEL and PSPCFGx pin states to be relatched. Therefore, the currently selected controller interface remains available after issuing a System Reset and waiting the required 25 μs.

FIGURE 7-1: ON-CHIP RESET CIRCUIT



7.3 Transmit Only Reset

A Transmit Only Reset is performed by setting the TXRST bit (ECON2<6>). The transmit logic is held in Reset until the bit is cleared. Any pending transmission is aborted and TXRTS (ECON1<1>) is cleared. To resume normal operation, clear the TXRST bit.

Both the POR and System Resets automatically perform a Transmit Reset, so this step does not need to be performed after a System or Power-on Reset. Only the transmit logic is affected by this operation. Other register and control blocks are not affected by this event.

7.4 Receive Only Reset

A Receive Only Reset is performed by setting the RXRST bit (ECON2<5>). The receive logic is held in Reset until the bit is cleared. Any packet being received is aborted and RXEN (ECON1<0>) is cleared. To resume normal operation, clear the RXRST bit.

Both the POR and System Resets automatically perform a Receive Reset, so this step does not need to be performed after a System or Power-on Reset. Only the receive logic is affected by this operation. Other register and control blocks are not affected by this event.

Following a Receive Only Reset, it is necessary to manually reconfigure the RX SFRs for normal receive operation again. For example, applications must clear the PKTCNT field in ESTAT by setting the PKTDEC bit (ECON1<8>) enough times for the count to reach zero. Similarly, applications must reset the ERXST and ERXTAIL Pointers before enabling reception again with the RXEN bit.

7.5 PHY Subsystem Reset

The PHY module may be reset by setting the PRST bit (PHCON1<15>). The PHY register contents all revert to their default values.

Unlike the Transmit and Receive Only Resets, the PHY cannot be removed from Reset immediately after setting PRST. The PHY requires a delay, after which the hardware automatically clears the PRST bit. It is recommended that, after issuing a Reset, the host controller polls PRST and waits for it to be cleared by hardware before using the PHY.

The POR and System Resets automatically perform a PHY Reset, so this step does not need to be performed after a System or Power-on Reset. Only the PHY is affected by this operation. Other register and control blocks are not affected by this event.

8.0 INITIALIZATION

Before using an ENCX24J600 device to transmit and receive packets, certain device settings must be initialized. Depending on the application, some configuration options may be left set to their default values. Those that need to be changed are typically set once after power-up and not changed thereafter.

8.1 Reset

Because it is possible for the host controller to reset independently from the ENCX24J600 (for example, when using an external debugger to reprogram the host), it is recommended that software issue a System Reset of the ENCX24J600 as the first step of its ordinary initialization routine.

Also, since it is possible for the host controller to exit its POR, begin code execution before the ENCX24J600 exits POR and latches the Interface mode, special care should be taken in the software to ensure that it does not attempt to blindly initialize the ENCX24J600 registers before the device is actually out of Reset. To take care of these potential pitfalls, it is recommended that firmware take a write-verify-reset-reverify approach to ensure proper start-up. For example:

1. Write 1234h to EUDAST.
2. Read EUDAST to see if it now equals 1234h. If it does not, the SPI/PSP interface may not be ready yet, so return to step 1 and try again.
3. Poll CLKRDY (ESTAT<12>) and wait for it to become set.
4. Issue a System Reset command by setting ETHRST (ECON2<4>).
5. In software, wait at least 25 μs for the Reset to take place and the SPI/PSP interface to begin operating again.
6. Read EUDAST to confirm that the System Reset took place. EUDAST should have reverted back to its Reset default of 0000h.
7. Wait at least 256 μs for the PHY registers and PHY status bits to become available.

The ENCX24J600 is now ready to accept further commands.

8.2 CLKOUT Frequency

If the ENCX24J600 is providing a system clock for the host controller, or other hardware features of the application, it is recommended that the application configure the output frequency on the CLKOUT pin first. The frequency is set by using the COCON<3:0> bits (ECON2<11:8>). By default, the output frequency on CLKOUT after a POR is 4 MHz. The last programmed frequency is maintained after all other Reset events.

For more information on using the output of the CLKOUT pin, see **Section 2.2 “CLKOUT Pin”**.

8.3 Receive Buffer

Before packet reception is enabled, the receive buffer must be configured by programming the ERXST Pointer. All memory between this pointer and the end of the physical memory (5FFFh), including those addresses, are reserved as the receive buffer for incoming packets. The value of ERXST must be word-aligned, since all incoming frames must be stored beginning at even addresses.

If an application expects a large amount of incoming traffic or frequent packet delivery, it is recommended that it allocate a larger receive buffer. Applications needing more space for saving old packets or other temporary storage, or wishing to hold several packets ready for transmission, can allocate less memory for the receive buffer.

Reception of incoming packets begins at the address designated by ERXST.

8.4 Transmit Buffer

No specific transmit buffer is defined. The host applications may write frames to be transmitted to any unused space in the SRAM buffer; no initialization is necessary.

8.5 Receive Filters

Before enabling packet reception, configure the receive filters to eliminate unwanted incoming packets. See **Section 10.0 “Receive Filters”** for details.

8.6 MAC Initialization

Once the receive buffer and filters are properly configured, several MAC registers must be configured. The order of programming is unimportant.

- If flow control operation is desired, configure the flow control module as described in **Section 11.0 “Flow Control”**.
- Verify that the TXCRCEN (MACON2<4>) and PADCFG<2:0> (MACON2<7:5>) bits are set correctly. Most applications will not need to modify these settings from their power-on defaults.
- Program the MAMXFL register with the maximum frame length to be accepted (received or transmitted). Most network nodes are configured to handle packets that are 1518 bytes or less (1522 bytes or less if VLAN tagging is used). Alternately, set HFRMEN (MACON2<2>) to accept any size frame.
- Set the RXEN bit (ECON1<0>) to enable packet reception by the MAC.

8.6.1 PREPROGRAMMED MAC ADDRESS

As shipped, each ENCX24J600 device has been preprogrammed with a unique MAC address. This value is stored in nonvolatile memory and reloaded into the MAADR registers after every Power-on and System Reset. The factory preprogrammed MAC address is permanent and will be restored to the MAC registers after each Reset.

The preprogrammed address in nonvolatile memory cannot be changed by the user, but it can be overwritten in the SFRs. If the user requires a different MAC address value, the MAADR registers will need to be written with the new MAC values by the host application after each Reset.

8.7 PHY Initialization

Depending on the application, the PHY may need to be configured during initialization. Typically, when using auto-negotiation, users should write 0x05E1 to PHANA to advertise flow control capability. Only special test code, such as when attempting to do loopback tests, needs other settings in the PHY to be reconfigured.

8.8 Other Considerations Following Reset

Beyond the steps already described, there are additional configuration options that may need to be adjusted following a device Reset. Normally, the default configurations of these items on Power-on Reset do not need to be changed.

For Half-Duplex mode:

- Verify that DEFER (MACON2<14>), BPEN (MACON2<13>) and NOBKOFF (MACON2<12>) are set correctly. These bits only apply when operating in Half-Duplex mode; most applications do not need to modify these settings from their power-on defaults. For IEEE 802.3 compliance, keep the DEFER bit set.
- Configure the Non-Back-to-Back Inter-Packet Gap register, MAIPG (Register 8-5). Most applications program this register to 12h, which selects maximum performance while complying with the IEEE 802.3 IPG previously specified.
- Set the MAXRET<3:0> (MACLCON<3:0>) bits to select the maximum number of retransmission attempts after a collision is detected. Most applications do not need to change this from the default value.

For Full-Duplex mode:

- Configure the low byte of the Non-Back-to-Back Inter-Packet Gap register, MAIPGL. Most applications program this register to 12h, which selects maximum performance while complying with the IEEE 802.3 IPG previously specified.

8.9 After Link Establishment

Several MAC configuration parameters are dependent upon the current duplex mode of the link. Once auto-negotiation completes, or the speed and duplex modes are manually reconfigured, these registers must be updated accordingly. For details about auto-negotiation and manual speed/duplex configuration, refer to **Section 12.0 "Speed/Duplex Configuration and Auto-Negotiation"**.

Once these steps are performed, packet reception is re-enabled by setting RXEN (ECON1<0>). The host controller may also begin to transmit packets as described in **Section 9.1 "Transmitting Packets"**. Before transmitting the first packet after link establishment or auto-negotiation, the MAC duplex configuration must be manually set to match the duplex configuration of the PHY. To do this, configure FULDPX (MACON2<0>) to match PHYDPX (ESTAT<10>).

For Half-Duplex mode, configure the Back-to-Back Inter-Packet Gap register, MABBIPG (Register 8-4), to set the nibble time offset delay between the end of one transmission and the beginning of the next in a back-to-back sequence. Program the register value as the desired period in nibble times, minus 6. Most applications will program this register to 12h, which represents the minimum Inter-Packet Gap (IPG) specified by IEEE 802.3, of 0.96 μs (at 100 Mb/s) or 9.6 μs (at 10 Mb/s).

For Full-Duplex mode, configure the Back-to-Back Inter-Packet Gap register, MABBIPG, to set the nibble time offset delay between the end of one transmission and the beginning of the next in a back-to-back sequence. The register value should be programmed as the desired period in nibble times, minus 3. Most applications will program this register to 15h, which represents the minimum IEEE 802.3 specified Inter-Packet Gap (IPG) of 0.96 μs (at 100 Mb/s) or 9.6 μs (at 10 Mb/s).

REGISTER 8-1: ECON2: ETHERNET CONTROL REGISTER 2

R/W-1	R/W-1	R/W-0	R/W-0	R/W-1(1)	R/W-0(1)	R/W-1(1)	R/W-1(1)
ETHEN	STRCH	TXMAC	SHA1MD5	COCON3	COCON2	COCON1	COCON0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
AUTOFC	TXRST	RXRST	ETHRST	MODLEN1	MODLEN0	AESLEN1	AESLEN0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **ETHEN:** Ethernet Enable bit
1 = Device is enabled (normal operation)
0 = Device is disabled (reduced power)

bit 14 **STRCH:** LED Stretching Enable bit
1 = Stretch transmit, receive and collision events on LEDA and LEDB to 50 ms
0 = LEDA and LEDB outputs show real-time status without stretching

bit 13 **TXMAC:** Automatically Transmit MAC Address Enable bit
1 = MAADR1-MAADR6 registers are automatically inserted into the source address field of all transmitted packets
0 = No automatic source address insertion

bit 12 **SHA1MD5:** SHA-1/MD5 Hash Control bit
1 = Hashing engine computes a SHA-1 hash
0 = Hashing engine computes an MD5 hash

bit 11-8 **COCON<3:0>:** CLKOUT Frequency Control bits(1)
1111 = 50 kHz nominal ((4 * FOSC)/2000)
1110 = 100 kHz nominal ((4 * FOSC)/1000)
1101 = No output (DC sinking to VSS)
1100 = 3.125 MHz nominal ((4 * FOSC)/32)
1011 = 4.000 MHz nominal ((4 * FOSC)/25)
1010 = 5.000 MHz nominal ((4 * FOSC)/20)
1001 = 6.250 MHz nominal ((4 * FOSC)/16)
1000 = 8.000 MHz nominal ((4 * FOSC)/12.5); duty cycle is not 50%
0111 = 8.333 MHz nominal ((4 * FOSC)/12)
0110 = 10.00 MHz nominal ((4 * FOSC)/10)
0101 = 12.50 MHz nominal ((4 * FOSC)/8)
0100 = 16.67 MHz nominal ((4 * FOSC)/6)
0011 = 20.00 MHz nominal ((4 * FOSC)/5)
0010 = 25.00 MHz nominal ((4 * FOSC)/4)
0001 = 33.33 MHz nominal ((4 * FOSC)/3)
0000 = No output (DC sinking to VSS)

bit 7 **AUTOFC:** Automatic Flow Control Enable bit
1 = Automatic flow control is enabled
0 = Automatic flow control is disabled

bit 6 **TXRST:** Transmit Logic Reset bit
1 = Transmit logic is held in Reset. TXRTS (ECON1<1>) is automatically cleared by hardware when this bit is set.
0 = Transmit logic is not in Reset (normal operation)

Note 1: Reset value on POR events only. All other Resets leave these bits unchanged.

REGISTER 8-1: ECON2: ETHERNET CONTROL REGISTER 2 (CONTINUED)

bit 6 **RXRST:** Receive Logic Reset bit
- 1 = Receive logic is held in Reset. RXEN (ECON1<0>) is automatically cleared by hardware when this bit is set.
- 0 = Receive logic is not in Reset (normal operation)

bit 4 **ETHRST:** Master Ethernet Reset bit
- 1 = All TX, RX, MAC, PHY, DMA, modular exponentiation, hashing and AES logic, and registers (excluding COCON) are reset. Hardware self-clears this bit to '0'. After setting this bit, wait at least 25 μs before attempting to read or write to the ENCX24J600 via the SPI or PSP interface.
- 0 = Device is not in Reset (normal operation)

bit 3-2 **MODLEN<1:0>:** Modular Exponentiation Length Control bits
- 11 = Reserved
- 10 = 1024-bit modulus and operands
- 01 = 768-bit modulus and operands
- 00 = 512-bit modulus and operands

bit 1-0 **AESLEN<1:0>:** AES Key Length Control bits
- 11 = Reserved
- 10 = 256-bit key
- 01 = 192-bit key
- 00 = 128-bit key

Note 1: Reset value on POR events only. All other Resets leave these bits unchanged.

REGISTER 8-2: EIDLED: ETHERNET ID STATUS/LED CONTROL REGISTER

R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0
LACFG3	LACFG2	LACFG1	LACFG0	LBCFG3	LBCFG2	LBCFG1	LBCFG0
bit 15							bit 8

R-0	R-0	R-1	R	R	R	R	R
DEVID2	DEVID1	DEVID0	REVID4	REVID3	REVID2	REVID1	REVID0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-12, bit 11-8 **LACFG<3:0>:** LEDA Configuration bits and **LBCFG<3:0>:** LEDB Configuration bits

- 1111 = Display link and speed state, transmit and receive events(1)
- 1110 = Display link and duplex state, transmit and receive events(1)
- 1101 = Reserved
- 1100 = Display link state, collision events; pin is driven high when a link is present and driven low temporarily when a collision occurs
- 1011 = Display link state, transmit and receive events; pin is driven high when a link is present and driven low while a packet is being received or transmitted
- 1010 = Display link state, receive events; pin is driven high when a link is present and driven low while a packet is being received
- 1001 = Display link state, transmit events; pin is driven high when a link is present and driven low while a packet is being transmitted
- 1000 = Display speed state; pin is driven high when in 100 Mbps mode and a link is present
- 0111 = Display duplex state; pin is driven high when the PHY is in full duplex (PHYDPX (ESTAT<10>) is ‘1’) and a link is present
- 0110 = Display transmit and receive events; pin is driven high while a packet is either being received or transmitted
- 0101 = Display receive events; pin is driven high while a packet is being received
- 0100 = Display transmit events; pin is driven high while a packet is being transmitted
- 0011 = Display collision events; pin is temporarily driven high when a collision occurs
- 0010 = Display link state; pin is driven high when linked
- 0001 = On (pin is driven high)
- 0000 = Off (pin is driven low)

bit 7-5 **DEVID<2:0>:** Device ID bits

001 = ENC624J600 family device

bit 4-0 **REVID<4:0>:** Silicon Revision ID bits

Indicates current silicon revision.

Note 1: These configurations require that a bi-color LED be connected between the LEDA and LEDB pins, and that LACFG<3:0> and LBCFG<3:0> be set to the same value. See **Section 2.5.1 “Using Bi-Color LEDs”** for detailed information.

REGISTER 8-3: MACON2: MAC CONTROL REGISTER 2

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
—	DEFER	BPEN	NOBKOFF	—	—	r	r
bit 15							bit 8

R/W-1	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-1	R/W-0
PADCFG2	PADCFG1	PADCFG0	TXCRCEN	PHDREN	HFRMEN	r	FULDPX
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14 **DEFER:** Defer Transmission Enable bit (applies to half duplex only)
- 1 = When the medium is occupied, the MAC will wait indefinitely for it to become free when attempting to transmit (use this setting for IEEE 802.3 compliance)
- 0 = When the medium is occupied, the MAC will abort the transmission after the excessive deferral limit is reached (24,288 bit times)

bit 13 **BPEN:** No Backoff During Back Pressure Enable bit (applies to half duplex only)
- 1 = After incidentally causing a collision during back pressure, the MAC immediately begins retransmitting
- 0 = After incidentally causing a collision during backpressure, the MAC delays using the binary exponential backoff algorithm before attempting to retransmit (normal operation)

bit 12 **NOBKOFF:** No Backoff Enable bit (applies to half duplex only)
- 1 = After any collision, the MAC immediately begins retransmitting
- 0 = After any collision, the MAC delays using the binary exponential backoff algorithm before attempting to retransmit (normal operation)

bit 11-10 **Unimplemented:** Read as '0'

bit 9-8 **Reserved:** Write as '0'

bit 7-5 **PADCFG<2:0>:** Automatic Pad and CRC Configuration bits
- 111 = All short frames are zero-padded to 64 bytes and a valid CRC is then appended
- 110 = No automatic padding of short frames
- 101 = MAC automatically detects VLAN protocol frames which have a 8100h type field and automatically pad to 64 bytes. If the frame is not a VLAN frame, it will be padded to 60 bytes. After padding, a valid CRC is appended.
- 100 = No automatic padding of short frames
- 011 = All short frames are zero-padded to 64 bytes and a valid CRC is then appended
- 010 = No automatic padding of short frames
- 001 = All short frames will be zero-padded to 60 bytes and a valid CRC is then appended
- 000 = No automatic padding of short frames

bit 4 **TXCRCEN:** Transmit CRC Enable bit
- 1 = MAC appends a valid CRC to all frames transmitted regardless of the PADCFG bits. TXCRCEN must be set if the PADCFG bits specify that a valid CRC will be appended.
- 0 = MAC does not append a CRC. The last 4 bytes are checked and if it is an invalid CRC, it is to be reported by setting CRCBAD (ETXSTAT<4>).

bit 3 **PHDREN:** Proprietary Header Enable bit
- 1 = Frames presented to the MAC contain a 4-byte proprietary header which is not used when calculating the CRC
- 0 = No proprietary header is present; the CRC covers all data (normal operation)

REGISTER 8-3: MACON2: MAC CONTROL REGISTER 2 (CONTINUED)

bit 2 **HFRMEN:** Huge Frame Enable bit
1 = Frames of any size will be allowed to be transmitted and received
0 = Frames bigger than MAMXFL will be aborted when transmitted or received

bit 1 **Reserved:** Write as '1'

bit 0 **FULDPX:** MAC Full-Duplex Enable bit
1 = MAC operates in Full-Duplex mode. For proper operation, the PHY must also be set to Full-Duplex mode.
0 = MAC operates in Half-Duplex mode. For proper operation, the PHY must also be set to Half-Duplex mode.

REGISTER 8-4: MABBIPG: MAC BACK-TO-BACK INTER-PACKET GAP REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-1	R/W-0
—	BBIPG6	BBIPG5	BBIPG4	BBIPG3	BBIPG2	BBIPG1	BBIPG0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-7 **Unimplemented:** Read as '0'

bit 6-0 **BBIPG<6:0>:** Back-to-Back Inter-Packet Gap Delay Time Control bits

When FULDPX (MACON2<0>) = 1:

Nibble time offset delay between the end of one transmission and the beginning of the next in a back-to-back sequence. The register value should be programmed to the desired period in nibble times minus 3. The recommended setting is 15h which represents the minimum IEEE specified Inter-Packet Gap (IPG) of 0.96 μs (at 100 Mb/s) or 9.6 μs (at 10 Mb/s).

When FULDPX (MACON2<0>) = 0:

Nibble time offset delay between the end of one transmission and the beginning of the next in a back-to-back sequence. The register value should be programmed to the desired period in nibble times minus 6. The recommended setting is 12h which represents the minimum IEEE specified Inter-Packet Gap (IPG) of 0.96 μs (at 100 Mb/s) or 9.6 μs (at 10 Mb/s).

REGISTER 8-5: MAIPG: MAC INTER-PACKET GAP REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0
—	r	r	r	r	r	r	r
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-1	R/W-0
—	IPG6	IPG5	IPG4	IPG3	IPG2	IPG1	IPG0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as ‘0’

bit 14-8 **Reserved:** Write as ‘0001100’ (0Ch)

bit 7 **Unimplemented:** Read as ‘0’

bit 6-0 **IPG<6:0>:** Non Back-to-Back Inter-Packet Gap Delay Time Control bits

Inter-Packet Gap (IPG) between the end of one packet received or transmitted and the start of the next packet transmitted. For maximum performance while meeting IEEE 802.3 compliance, leave this field set to 12h, which represents an Inter-Packet Gap time of 0.96 μs (at 100 Mb/s) or 9.6 μs (at 10 Mb/s).

REGISTER 8-6: MACLCON: MAC COLISION CONTROL REGISTER

U-0	U-0	R/W-1	R/W-1	R/W-0	R/W-1	R/W-1	R/W-1
—	—	r	r	r	r	r	r
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	—	MAXRET3	MAXRET2	MAXRET1	MAXRET0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **Reserved:** Write as ‘110111’ (37h)

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **MAXRET<3:0>:** Maximum Retransmissions Control bits (half duplex only)

Maximum retransmission attempts the MAC will make before aborting a packet due to excessive collisions.

9.0 TRANSMITTING AND RECEIVING PACKETS

Beyond providing the transceiver interface to the network medium, ENC424J600/624J600 devices also handle many of the mechanical tasks of packet management, off-loading much of the routine Ethernet housekeeping from the host application. The device manages the separate transmit and receive buffers, handles transmission and potential collisions, filters incoming packets, and stores received packets with the additional information required for processing. The host controller writes data to the memory, configures the length of the packet to send, initiates the transmissions and reads incoming packets from the receive buffer. Padding and checksum generation, as well as status information on received packets, are all handled automatically.

9.1 Transmitting Packets

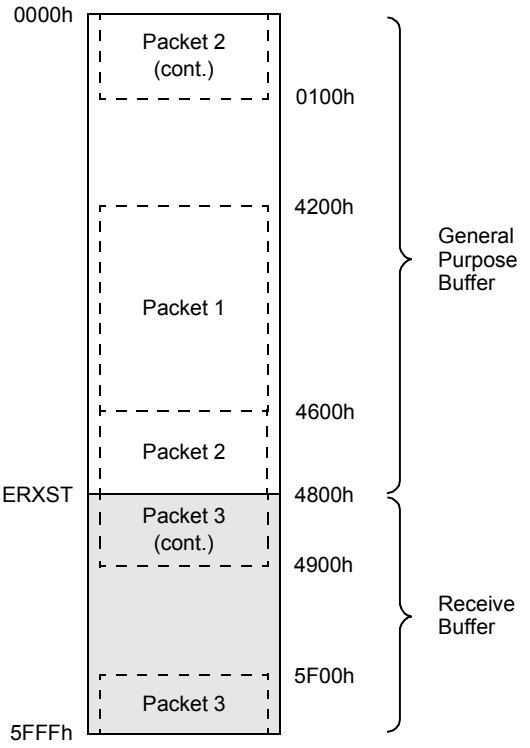
The general purpose buffer is bounded by the beginning of the address space (0000h) and the last byte before the beginning of the receive buffer (ERXST – 1). Since ERXST must be word-aligned, both buffers start on even addresses and end on odd addresses. For details on buffer allocation, see **Section 3.5 "SRAM Buffer"**.

The packet to be transmitted is defined by two values: the Transmit Data Start Pointer, ETXST, and the Transmit Buffer Length Pointer, ETXLEN. When transmitting a packet, the device reads the ETXLEN bytes, beginning at the address indicated by ETXST. If the end of the general purpose buffer is encountered during this process, the operation will wrap around to the beginning of the general purpose buffer space (0000h). Packets can also be transmitted directly from the receive buffer (for instance, when changing the source and destination addresses). If the end of the receive buffer is encountered, the operation wraps to the beginning of the receive buffer instead. This wrap-around behavior precludes packets from spanning both buffers.

Figure 9-1 shows three examples of the wrapping behavior. Packet 1 in the diagram is transmitted without any wrapping. Packet 2 reaches the end of the general purpose buffer, and therefore, wraps to address 0000h. Packet 3 is being transmitted from the receive buffer, and therefore, wraps to ERXST when the end of the receive buffer is reached.

The device can be configured to insert the source MAC address using the values from the MAADR registers. This feature is enabled by setting the TXMAC bit (ECON2<13>). When enabled, the device reads the 6-byte destination address from memory, inserts the 6-byte source MAC address from the MAADR registers into the transmitted byte stream, then continues reading and transmitting the remaining bytes from memory.

FIGURE 9-1: EXAMPLES OF TX BUFFER WRAPPING



Example Packet	ETXST	ETXLEN
1	4200h	1024
2	4600h	768
3	5F00h	512

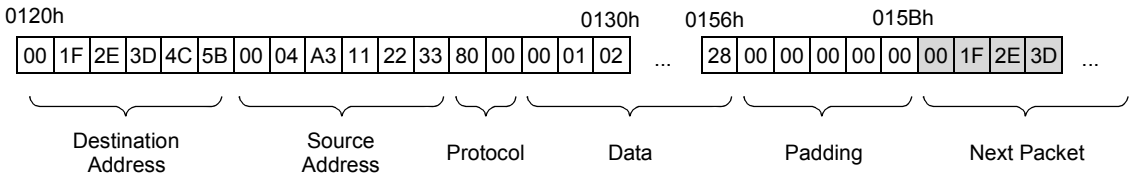
The value of ETXLEN only indicates the number of bytes to read from memory, *not* the number of bytes to be transmitted. If the device is configured to insert the source MAC address, add padding or append the CRC; the actual number of bytes transmitted on the physical medium will increase. Figure 9-2 shows how to configure ETXLEN for three identical packets of data when various transmission options are configured.

Before transmitting any packets, the device needs to be initialized (see **Section 8.0 "Initialization"**). Setting TXRTS (ECON1<1>) initiates the transmission. This bit is automatically cleared by hardware when the operation is complete. In addition, the device can also be configured to assert the TXIF interrupt and the external interrupt signal on completion (see **Section 13.0 "Interrupts"** for additional details).

Transmission operations can be aborted by manually clearing the TXRTS bit at any time. If a packet transmission is in progress, it will be aborted immediately and the device will send a jam signal, effectively notifying the link partner to discard any partial packet it has received.

FIGURE 9-2: EXAMPLES FOR SELECTING ETXLEN VALUES

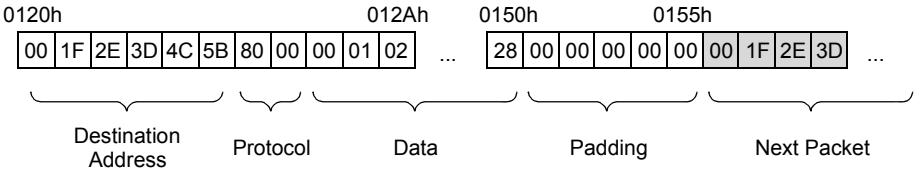
Example 1: Source Address and Padding Provided by Application



ETXST = 0120h
ETXLEN = **3Ch**

MAC address insertion disabled (TXMAC = 0)
Automatic padding disabled (PADCFG<2:0> = 000)
CRC generation enabled (TXCRCEN = 1)

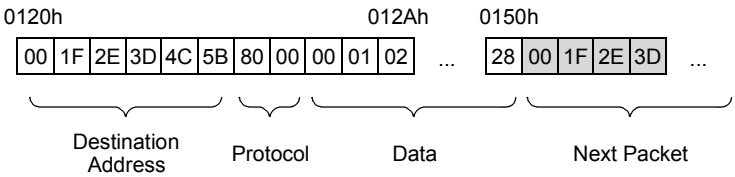
Example 2: Padding Provided by Application, Source Address to be Inserted by ENCX24J600



ETXST = 0120h
ETXLEN = **36h**

MAC address insertion enabled (TXMAC = 1)
Automatic padding disabled (PADCFG<2:0> = 000)
CRC generation enabled (TXCRCEN = 1)

Example 3: Source Address and Padding to be Inserted by ENCX24J600



ETXST = 0120h
ETXLEN = **31h**

MAC insertion enabled (TXMAC = 1)
Automatic padding enabled (PADCFG<2:0> = 101)
CRC generation enabled (TXCRCEN = 1)

While transmission is active (TXRTS is set), it is recommended that ETXST and ETXLEN, as well as the TXMAC bit (ECON2<13>), not be modified. Since ERXST controls the end of the transmit buffer, and therefore, buffer wrap-around, it must also remain unchanged.

To transmit a packet:

1. Initialize the MAC as described in **Section 8.6 "MAC Initialization"**. Most applications should leave PADCFG<3:0> and TXCRCEN set to their default values, which enables automatic padding and CRC generation. For automatic insertion of the source MAC address during transmission, set the TXMAC bit to '1'.
2. If desired, enable the transmit done and/or transmit abort interrupts by setting TXIE and/or TXABTIE (EIE<3:2>). Clear TXIF and TXABTIF (EIR<3:2>) if they are currently set. To generate the interrupt, also set INTIE (EIE<15>).
3. Copy the packet to the SRAM buffer.
4. Program ETXST to the start address of the packet.
5. Program ETXLEN with the length of data copied to the memory.
6. Set the TXRTS bit to initiate transmission.
7. Wait for the hardware to clear TXRTS and trigger a transmit interrupt, indicating transmission has completed.
8. Read the ETXSTAT register for status information as described in the next section.

The transmit function does not modify the ETXST Pointer or ETXLEN data length after the operation completes. To send another packet, the Start Pointer must be manually moved to the location of the next packet and the transmit length must be updated. If desired, the application can retransmit the last packet by setting TXRTS again without modifying ETXST or ETXLEN.

9.1.1 TRANSMISSION STATUS

After transmitting a packet (either successfully or unsuccessfully), the ETXSTAT and ETXWIRE registers contain status information about the operation. The values in these registers will persist until the next packet is transmitted (again, either successfully or unsuccessfully). Therefore, ETXSTAT and ETXWIRE should be treated as valid only when TXRTS is clear.

The LATECOL (ETXSTAT<10>), MAXCOL (ETXSTAT<9>) and EXDEFER (ETXSTAT<8>) bits are error flags indicating that packet transmission has failed. (These errors are possible only in Half-Duplex mode; therefore, these status bits should be ignored when operating in Full-Duplex mode.) The device asserts these flags and clears the TXRTS bit to prevent a single packet from stalling device operation. When any of these flags are set, the packet was not successfully transmitted and the host controller should determine whether to retry or ignore the error.

The CRCBAD (EXTSTAT<4>) bit is a warning. It is only meaningful when automatic CRC generation is disabled and indicates that the checksum computed by the MAC did not match the one appended by software. If the software CRC is incorrect, the packet will be rejected by the remote node. When automatic MAC hardware generation of the CRC is enabled, this bit can be ignored as the CRC is always correct.

The DEFER bit (ETXSTAT<7>) and the COLCNT<3:0> bits (ETXSTAT<3:0>) are status indicators. DEFER simply indicates that the device had to wait before transmitting due to flow control or other traffic on the network. The COLCNT bits indicate the number of collisions that occurred before the packet was successfully transmitted.

The ETXWIRE register is a count of the number of actual bytes the MAC transmitted onto the physical medium before the transmission completed, either successfully or unsuccessfully. In Full-Duplex mode, this count is the total length of the packet, including padding and CRC. In Half-Duplex mode, this status register includes all extra bytes that were transmitted due to any collisions that occurred. Therefore, it can be used to gauge how much total bandwidth the application is using.

9.1.2 SPECIAL CASE TRANSMISSION

When the value of ETXLEN is 07h or less, the ability to set the TXRTS bit is locked out in hardware. This is because the resulting packet would be unable to meet IEEE 802.3 requirements.

If the PHY is unlinked at the time software sets the TXRTS bit to transmit a packet, the transmission will complete normally with applicable interrupts still occurring. However, the PHY submodule will also suppress the transmission of any data onto the physical medium. This avoids interference with auto-negotiation, which may be already using the physical medium. This behavior is also necessary to meet IEEE 802.3 specifications.

If an attempt is made to transmit a packet that is larger than specified in the MAC Maximum Frame Length register, and huge frames are disabled (MACON2<2> = 0), the transmission will start normally. However, once the MAC has transmitted the number of bytes defined in MAMXFL, the MAC will immediately cease transmission. This results in the packet being partially transmitted and then truncated without a valid CRC being appended. In almost all cases, this results in the remote node rejecting the packet as having an invalid CRC.

In full duplex, the MAC inhibits transmission of any packets until the pause timer expires when two conditions are met:

- Flow control is enabled (RXPAUS bit is set) and
- A valid pause frame was received from the remote node

It will still be possible for software to set the TXRTS bit to start a transmission. However, this has the effect of queuing the packet for future transmission instead of causing an immediate transmission to start. Once the pause timer expires, the queued packet will transmit normally, causing any applicable interrupts to occur.

9.2 Receiving Packets

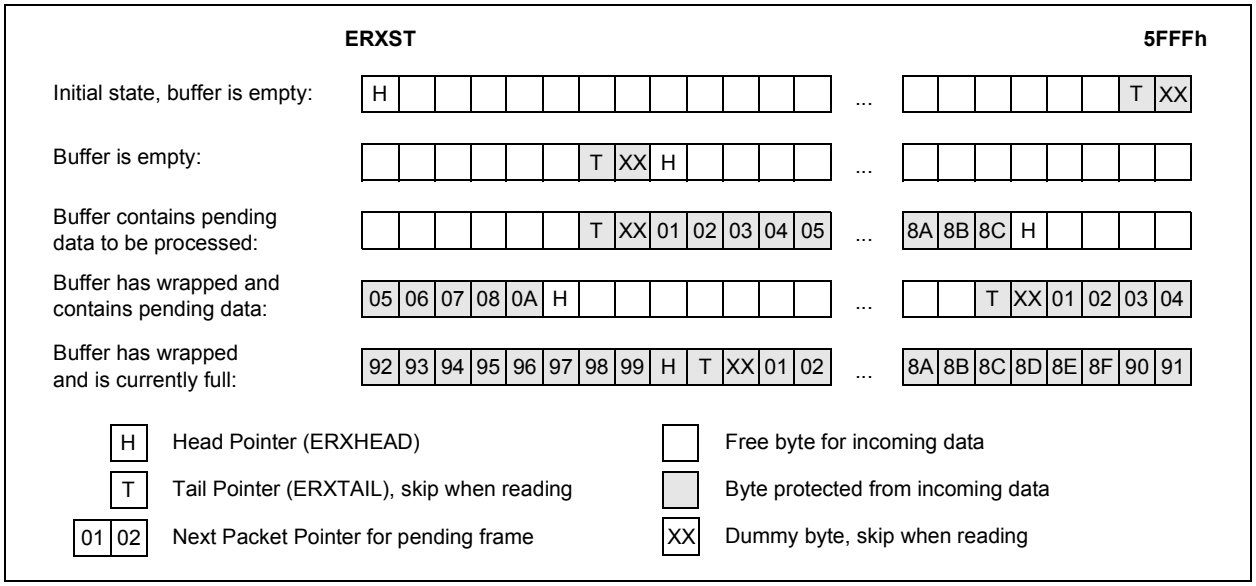
As Ethernet frames arrive, they are written to the circular receive buffer, bounded by the Receive Buffer Start Address (ERXST) register and the end of the physical memory at 5FFFh. The hardware also maintains a counter indicating the number of pending frames.

Each frame starts on an even address. The hardware maintains a Receive Head Pointer, ERXHEAD, indicating the next location to be written, and automatically wraps back to ERXST when it reaches the end of memory. The Tail Pointer, ERXTAIL, is maintained by software. Addresses from the Tail Pointer, up to the Head Pointer, are considered to be protected by software. This allows the host controller to prevent incoming frames from overwriting data that has not yet been processed.

When ERXTAIL points to the same location as ERXHEAD, the receive packet buffer is considered to be full. Due to this definition, there is no empty condition. For simplicity, applications may choose to keep the Tail Pointer always set to two bytes behind the next frame to be processed, or two bytes behind the Head Pointer when no frames are pending. Figure 9-3 shows these pointer relationships.

If ERXHEAD reaches ERXTAIL while receiving a frame, or if the receive filters reject the packet, the ERXHEAD Pointer is rolled back to its previous location and the packet is discarded.

FIGURE 9-3: EXAMPLES OF RECEIVE BUFFER WRAP BETWEEN ERXHEAD AND ERXTAIL



It is possible for the host application to write to the receive buffer. However, it is recommended not to do so outside of the area protected by the Tail Pointer in order to prevent it from being subsequently overwritten by future receive packets.

ERXHEAD is a read-only register and may be updated at any time by hardware. The high byte is shadowed to ensure it can be safely read on 8-bit interfaces (SPI or PSP). When reading ERXHEAD, read the low byte first. This operation simultaneously copies the high byte to a shadow register. Reading the high byte automatically reads from this shadow register. This ensures that the value has not been modified since the low byte was obtained, even if another packet has been received in the interim.

9.2.1 CONFIGURING PACKET RECEPTION

Once the MAC and PHY are properly initialized, the device is ready to begin receiving packets.

To enable reception:

1. Program the ERXST Pointer (low byte first if writing a byte at a time) to the first address to be used for the receive buffer. This pointer must indicate an even address. The Head Pointer, ERXHEAD, will automatically be set to the same value.
2. In the host controller application, create a variable, `NextPacketPointer`, to hold the address value of the next received packet. Initialize this variable to be equal to the current value of ERXST.
3. Program the Tail Pointer, ERXTAIL, to the last even address of the buffer or 5FFEh.
4. Configure interrupts as desired. See **Section 13.0 "Interrupts"** for more information.
5. Set RXEN (ECON1<0>) to enable reception.

Once RXEN is set, it is recommended that ERXST not be modified. The host controller must monitor the ENCX24J600 to determine when a packet has arrived and is ready to be processed. This is accomplished by using the packet pending interrupt as described in **Section 13.1.5 "Received Packet Pending"**. Alternatively, poll the PKTCNT bits for a non-zero value.

9.2.2 STORAGE OF INCOMING PACKETS

Packets are stored sequentially in the receive buffer. Each frame is stored as it was presented to the MAC, including all padding and frame check (CRC) bytes, but excluding any preamble or start of stream/frame delimiter bytes. Frames are always saved starting on an even address, so those with an odd length skip one byte before the next frame begins. A sample packet stored in memory is shown in Figure 9-4.

When a packet is received, the hardware increments the Packet Counter bits, PKTCNT (ESTAT<7:0>). Incoming bytes are written sequentially, beginning at the Head Pointer, ERXHEAD. If the Head Pointer reaches the Tail Pointer, ERXTAIL, during reception, or if incrementing the PKTCNT bits would cause an overflow, the packet will be discarded and the Head Pointer restored.

Each received frame is preceded in memory by a pointer to the next frame and a Receive Status Vector (RSV). The RSV includes the length of the frame, and flags indicating the type of packet and which filters were matched. This format of the RSV is shown in Table 9-1.

To retrieve a packet from the buffer:

1. Verify that a packet is waiting by ensuring that the PKTCNT<7:0> bits are non-zero or that PKTIF (EIR<6>) is set.
2. Begin reading at address pointed to by the application variable, `NextPacketPointer` (see **Section 9.2.1 "Configuring Packet Reception"**).
3. Read the first two bytes of the packet, which are the address of the next packet and write to `NextPacketPointer`.
4. Read the next six bytes, which are the Receive Status Vector (RSV).
5. Read the Ethernet frame. The number of bytes to be read is indicated by the received byte count in the RSV read during step 4.
6. As the frame is read and processed, incremental amounts of memory buffer can be freed up by updating the ERXTAIL Pointer value to the point where the packet has been processed, taking care to wrap back at the end of the received memory buffer. Once the whole frame has been processed, the final value of ERXTAIL should be equal to (`NextPacketPointer` – 2).
7. Set PKTDEC (ECON1<8>) to decrement the PKTCNT bits. PKTDEC is automatically cleared by hardware if PKTCNT decrements to zero.

FIGURE 9-4: EXAMPLE OF A RECEIVED PACKET IN BUFFER MEMORY

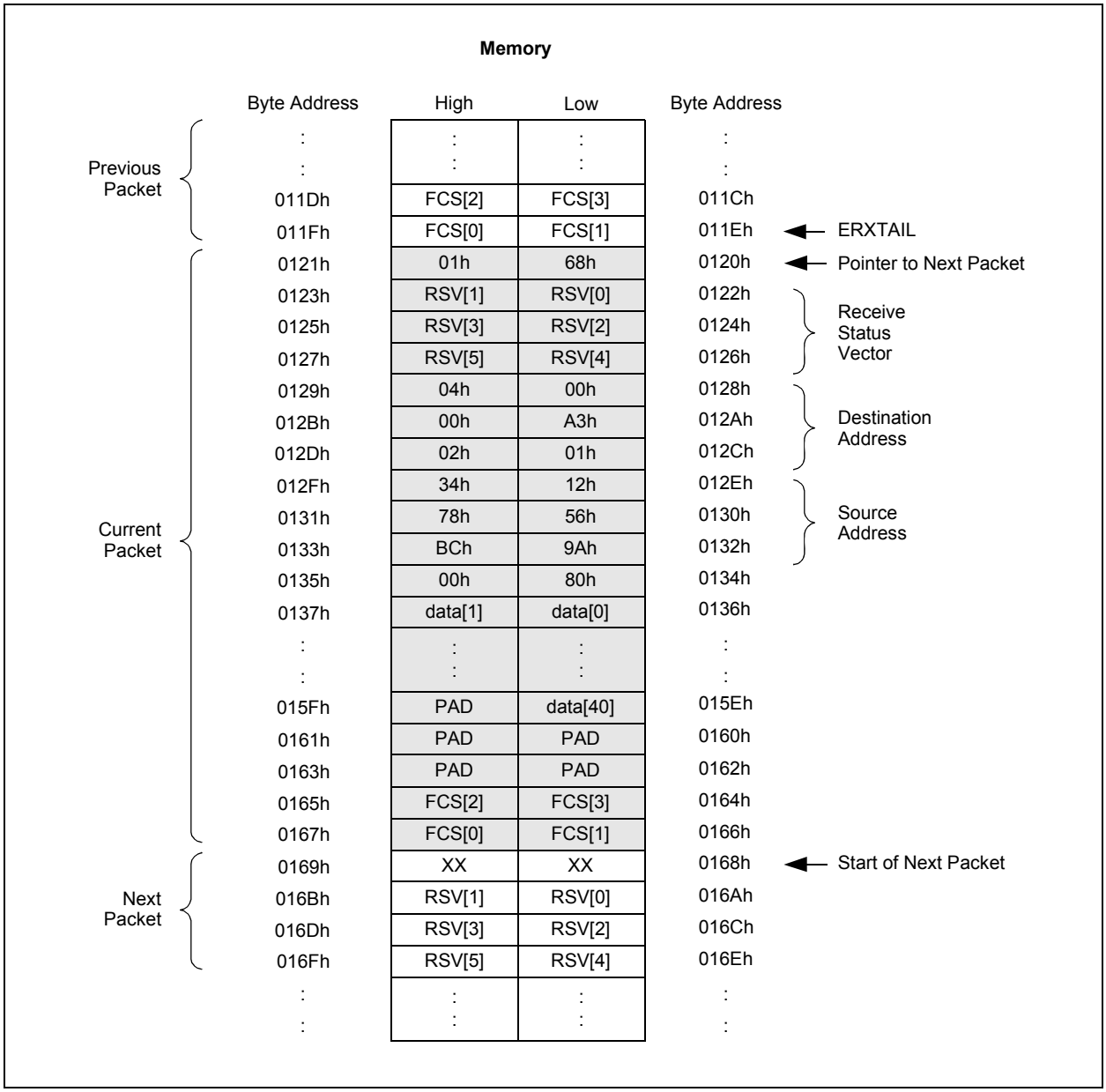


TABLE 9-1: RECEIVE STATUS VECTOR

Byte	Bit(s)	Field	Description
5	47:40	Zeros	00h
4	39	Zero	‘0’
	38	Reserved	
	37	Reserved	
	36	Unicast Filter Match	Current frame met criteria for the Unicast Receive filter.
	35	Pattern Match Filter Match	Current frame met criteria for the Pattern Match Receive filter as configured when the packet was received.
	34	Magic Packet™ Filter Match	Current frame met criteria for the Magic Packet Receive filter as configured when the packet was received.
	33	Hash Filter Match	Current frame met criteria for the Hash Receive filter as configured when the packet was received.
	32	Not-Me Filter Match	Current frame met criteria for the Not-Me Receive filter.
3	31	Runt Filter Match	Current frame met criteria for the Runt Packet Receive filter.
	30	Receive VLAN Type Detected	Current frame was recognized as a VLAN tagged frame.
	29	Receive Unknown Opcode	Current frame was recognized as a control frame but it contained an unknown opcode.
	28	Receive Pause Control Frame	Current frame was recognized as a control frame containing a valid pause frame opcode and a valid destination address.
	27	Receive Control Frame	Current frame was recognized as a control frame for having a valid type/length designating it as a control frame.
	26	Dribble Nibble	Indicates that after the end of this packet, an additional 1 to 7 bits were received. The extra bits were thrown away.
	25	Receive Broadcast Packet	Current frame has a valid Broadcast address.
	24	Receive Multicast Packet	Current frame has a valid Multicast address.
2	23	Received Ok	Received packet had a valid CRC and no symbol errors.
	22	Length Out of Range	Frame type/length field was larger than 1500 bytes (type field).
	21	Length Check Error	Frame length field value in the packet does not match the actual data byte length and specifies a valid length.
	20	CRC Error	Frame CRC field value does not match the CRC calculated by the MAC.
	19	Reserved	
	18	Carrier Event Previously Seen	A carrier event was detected at some time since the last receive. The carrier event is not associated with this packet. A carrier event is activity on the receive channel that does not result in a packet receive attempt being made.
	17	Reserved	
	16	Packet Previously Ignored	A frame larger than 50,000 bit times occurred or a packet has been dropped since the last receive.
1	15:0	Received Byte Count	Length of the received frame in bytes. This includes the destination address, source address, type/length, data, padding and CRC fields. This field is stored in little-endian format.
0			

REGISTER 9-1: ECON1: ETHERNET CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MODEXST	HASHEN	HASHOP	HASHLST	AESST	AESOP1	AESOP0	PKTDEC
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FCOP1	FCOP0	DMAST	DMACPY	DMACSSD	DMANOCS	TXRTS	RXEN
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **MODEXST:** Modular Exponentiation Start bit
- 1 = Modular exponentiation calculation started/busy; automatically cleared by hardware when done
- 0 = Modular exponentiation calculation done/Idle

bit 14 **HASHEN:** MD5/SHA-1 Hash Enable bit
- 1 = MD5/SHA-1 hashing engine enabled. Data written to the hashing engine by the DMA is added to the hash.
- 0 = MD5/SHA-1 hashing engine disabled

bit 13 **HASHOP:** MD5/SHA-1 Hash Operation Control bit
- 1 = MD5/SHA-1 hash engine loads the Initial Value (IV) from the hash memory. This mode is typically used for HMAC hash operations.
- 0 = Normal MD5/SHA-1 hash operation

bit 12 **HASHLST:** MD5/SHA-1 Hash Last Block Control bit
- 1 = The next DMA transfer to the hash engine completes the hash. If needed, padding is automatically generated and added to the hash.
- 0 = The next DMA transfer to the hash engine adds data to the hash. Further data additions to the hash are still possible.

bit 11 **AESST:** AES Encrypt/Decrypt Start bit
- 1 = AES encrypt/decrypt operation is started/busy; automatically cleared by hardware when done
- 0 = AES encrypt/decrypt operation is done/Idle

bit 10-9 **AESOP<1:0>:** AES Operation Control bits
- 11 = Reserved
- 10 = ECB/CBC decrypt
- 01 = CBC/CFB encrypt
- 00 = ECB/CFB/OFB encrypt or key initialization

bit 8 **PKTDEC:** RX Packet Counter Decrement Control bit
- 1 = Decrement PKTCNT (ESTAT<7:0>) bits by one. Hardware immediately clears PKTDEC to '0', allowing back-to-back decrement operations.
- 0 = Leave PKTCNT bits unchanged

bit 7-6 **FCOP<1:0>:** Flow Control Operation Control/Status bits

When FULDPX (MACON2<0>) = 1:
- 11 = End flow control by sending a pause frame with 0000h pause timer value; automatically cleared by hardware when done
- 10 = Enable flow control by periodically sending pause frames with a pause timer defined by EPAUS
- 01 = Transmit single pause frame defined by EPAUS; automatically cleared by hardware when done
- 00 = Flow control disabled/Idle

When FULDPX (MACON2<0>) = 0:
- 1x, 01 = Enable flow control by continuously asserting backpressure (transmitting preamble)
- 00 = Flow control disabled/Idle

REGISTER 9-1: ECON1: ETHERNET CONTROL REGISTER 1 (CONTINUED)

bit 5 **DMAST:** DMA Start bit
1 = DMA is started/busy; automatically cleared by hardware when done
0 = DMA is done/Idle

bit 4 **DMACPY:** DMA Copy Control bit
1 = DMA copies data to memory location at EDMADST
0 = DMA does not copy data; EDMADST is ignored

bit 3 **DMACSSD:** DMA Checksum Seed Control bit
1 = DMA checksum operations are initially seeded by the one's complement of the checksum contained in EDMACS
0 = DMA checksum operations are initially seeded by 0000h

bit 2 **DMANOCS:** DMA No Checksum Control bit
1 = DMA does not compute checksums; EDMACS remains unchanged
0 = DMA computes checksums; hardware updates EDMACS at the completion of all DMA operations

bit 1 **TXRTS:** Transmit Request to Send Status/Control bit
1 = Transmit an Ethernet frame; automatically cleared by hardware when done
0 = Transmit logic done/Idle

bit 0 **RXEN:** Receive Enable bit
1 = Packets which pass the current RX filter configuration are written to the receive buffer
0 = All packets received are ignored

REGISTER 9-2: ETXSTAT: ETHERNET TRANSMIT STATUS REGISTER

U-0	U-0	U-0	R-0	R-0	R-0	R-0	R-0
—	—	—	r	r	LATECOL(1)	MAXCOL(1)	EXDEFER(1)
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
DEFER(1)	r	r	CRCBAD	COLCNT3(1)	COLCNT2(1)	COLCNT1(1)	COLCNT0(1)
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-13 **Unimplemented:** Read as '0'

bit 12-11 **Reserved:** Ignore on read

bit 10 **LATECOL:** Transmit Late Collision Status bit(1)
- 1 = A collision occurred after transmitting more than MACLCONH + 8 bytes. The last transmission was aborted.
- 0 = No late collision occurred during the last transmission

bit 9 **MAXCOL:** Transmit Maximum Collisions Status bit(1)
- 1 = MACLCONL + 1 collisions occurred while transmitting the last packet. The last transmission was aborted.
- 0 = MACLCONL or less collisions occurred while transmitting the last packet

bit 8 **EXDEFER:** Transmit Excessive Defer Status bit(1)
- 1 = The medium was busy with traffic from other nodes for more than 24,288 bit times. The last transmission was aborted.
- 0 = The MAC deferred for less than 24,288 bit times while transmitting the last packet

bit 7 **DEFER:** Transmit Defer Status bit(1)
- 1 = The medium was busy with traffic from other nodes, so the MAC was forced to temporarily defer transmission of the last packet
- 0 = No transmit deferral or an excessive deferral occurred while attempting to transmit the last packet

bit 6-5 **Reserved:** Ignore on read

bit 4 **CRCBAD:** Transmit CRC Incorrect Status bit
- 1 = The FCS field of the last packet transmitted did not match the CRC internally generated by the MAC during transmission
- 0 = The FCS field of the last packet transmitted was correct or the MAC is configured to append an internally generated CRC

bit 3-0 **COLCNT<3:0>:** Transmit Collision Count Status bits(1)

Number of collisions that occurred while transmitting the last packet.

Note 1: Applicable in Half-Duplex mode only; collisions and deferrals are not possible in Full-Duplex mode.

REGISTER 9-3: ESTAT: ETHERNET STATUS REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
INT	FCIDLE	RXBUSY	CLKRDY	r	PHYDPX	r	PHYLNK
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PKTCNT7	PKTCNT6	PKTCNT5	PKTCNT4	PKTCNT3	PKTCNT2	PKTCNT1	PKTCNT0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **INT:** Interrupt Pending Status bit
- 1 = One of the EIR bits is set and enabled by the EIE register. If INTIE (EIE<15>) is set, the $\overline{INT}$ pin is also driven low.
- 0 = No enabled interrupts are currently pending. The $\overline{INT}$ pin is being driven high.

bit 14 **FCIDLE:** Flow Control Idle Status bit
- 1 = Internal flow control state machine is Idle. It is safe to change the FCOP (ECON1<7:6>) and FULDPX (MACON2<0>) bits.
- 0 = Internal flow control state machine is busy. Do not modify the FCOP (ECON1<7:6>) or FULDPX (MACON2<0>) bits.

bit 13 **RXBUSY:** Receive Logic Active Status bit
- 1 = Receive logic is currently receiving a packet. This packet may be discarded in the future if an RX buffer overflow occurs or a receive filter rejects it, so this bit does not necessarily indicate that an RX packet pending interrupt will occur.
- 0 = Receive logic is Idle

bit 12 **CLKRDY:** Clock Ready Status bit
- 1 = Normal operation
- 0 = Internal Ethernet clocks are not running and stable yet. Only the ESTAT and EUDAST registers should be accessed.

bit 11 **Reserved:** Ignore on read

bit 10 **PHYDPX:** PHY Full Duplex Status bit
- 1 = PHY is operating in Full-Duplex mode
- 0 = PHY is operating in Half-Duplex mode

bit 9 **Reserved:** Ignore on read

bit 8 **PHYLNK:** PHY Linked Status bit
- 1 = Ethernet link has been established with a remote Ethernet partner
- 0 = No Ethernet link present

bit 7-0 **PKTCNT<7:0>:** Receive Packet Count bits

Number of complete packets that are saved in the RX buffer and ready for software processing. Set the PKTDEC (ECON1<8>) bit to decrement this field.

NOTES:

10.0 RECEIVE FILTERS

To minimize the number of frames that the host controller must process, ENC424J600/624J600 devices incorporate 11 different receive filters to discard unwanted frames. The following filters are available:

- CRC Error Collection Filter
- Runt Error Collection Filter
- CRC Error Rejection Filter
- Runt Error Rejection Filter
- Unicast Collection Filter
- Not-Me Unicast Collection Filter
- Multicast Collection Filter
- Broadcast Collection Filter
- Hash Table Collection Filter
- Magic Packet™ Collection Filter
- Pattern Match Collection Filter

Each filter is software configurable, and can be individually enabled or disabled, using the ERXFCON register (Register 10-1). Each filter is either a Collection or a Rejection filter, with incoming frames passing sequentially through each enabled filter. The first filter to make a definitive decision for a frame takes priority over all others. Collection filters either force a frame to be accepted or defer the decision to a lower priority filter. Similarly, Rejection filters either discard frames or defer to lower priority filters. Frames that pass through all filters without specifically being accepted are discarded. Figure 10-1 demonstrates this decision tree.

At power-up, the CRC Error Rejection, Runt Error Rejection, Unicast Collection and Broadcast Collection filters are enabled, and all others are disabled. With these settings, the device will only accept Broadcast frames and frames specifically addressed to the local MAC address. Invalid frames and those destined for other nodes will be automatically rejected.

Note 1: The MAC internally processes and filters Ethernet control frames as they arrive and before they reach these filters. For the application to receive Ethernet control frames, enable the PASSALL option (MACCON1<1> = 1).

2: If the Ethernet Receive Enable bit, RXEN (ECON1<0>), is set, the filters may make an incorrect decision if any of the receive filters are reconfigured at the exact moment a new frame is being received. To avoid this behavior, clear the RXEN bit prior to changing any receive filter settings.

REGISTER 10-1: ERXFCON: ETHERNET RX FILTER CONTROL REGISTER

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
HTEN	MPEN	—	NOTPM	PMEN3	PMEN2	PMEN1	PMEN0
bit 15							bit 8

R/W-0	R/W-1	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-1
CRCEEN	CRCEN	RUNTEEN	RUNTEN	UCEN	NOTMEEN	MCEN	BCEN
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **HTEN:** Hash Table Collection Filter Enable bit
1 = Accept packets having a hashed destination address that points to a set bit in the Hash Table(1)
0 = Filter is disabled

bit 14 **MPEN:** Magic Packet™ Collection Filter Enable bit
1 = Accept packets containing a Magic Packet pattern for the local MAC address(1)
0 = Filter is disabled

bit 13 **Unimplemented:** Read as '0'

bit 12 **NOTPM:** Pattern Match Inversion Control bit
1 = Pattern Match checksum mismatch required for a successful Pattern Match
0 = Pattern Match checksum match required for a successful Pattern Match

bit 11-8 **PMEN<3:0>:** Pattern Match Collection Filter Enable bits
When NOTPM = 0:
A packet is accepted by the filter if the pattern checksum matches AND the selected mode's condition is true.
When NOTPM = 1:
A packet is accepted by the filter if pattern checksum does *not* match AND the selected mode's condition is true.
1111
.... = Reserved
1010
1001 = Magic Packet for local Unicast address(1)
1000 = Hashed packet destination points to a bit in the Hash Table registers that is set(1)
0111 = Packet destination is *not* the Broadcast address(1)
0110 = Packet destination is the Broadcast address(1)
0101 = Packet destination is *not* a Multicast address(1)
0100 = Packet destination is a Multicast address(1)
0011 = Packet destination is *not* the local Unicast address(1)
0010 = Packet destination is the local Unicast address(1)
0001 = Accept all packets with a checksum match defined by NOTPM(1)
0000 = Filter is disabled

bit 7 **CRCEEN:** CRC Error Collection Filter Enable bit
1 = Packets with an invalid CRC will be accepted, regardless of all other filter settings
0 = Filter is disabled

Note 1: This filtering decision can be overridden by the CRC Error Rejection filter and Runt Error Rejection filter decisions, if enabled, by CRCEN or RUNTEN.

2: This filtering decision can be overridden by the CRC Error Collection filter and Runt Error Collection filter decisions, if enabled, by CRCEEN or RUNTEEN.

REGISTER 10-1: ERXFCON: ETHERNET RX FILTER CONTROL REGISTER (CONTINUED)

bit 6 **CRCEN:** CRC Error Rejection Filter Enable bit
1 = Packets with an invalid CRC will be discarded(2)
0 = Filter is disabled

bit 5 **RUNTEEN:** Runt Error Collection Filter Enable bit
1 = Accept packets that are 63 bytes or smaller, regardless of all other filter settings
0 = Filter is disabled

bit 4 **RUNTEN:** Runt Error Rejection Filter Enable bit
1 = Discard packets that are 63 bytes or smaller(2)
0 = Filter is disabled

bit 3 **UCEN:** Unicast Destination Collection Filter Enable bit
1 = Accept packets with a destination address matching the local MAC address(1)
0 = Filter is disabled

bit 2 **NOTMEEN:** Not-Me Unicast Destination Collection Filter Enable bit
1 = Accept packets with a Unicast destination address that does not match the local MAC address(1)
0 = Filter is disabled

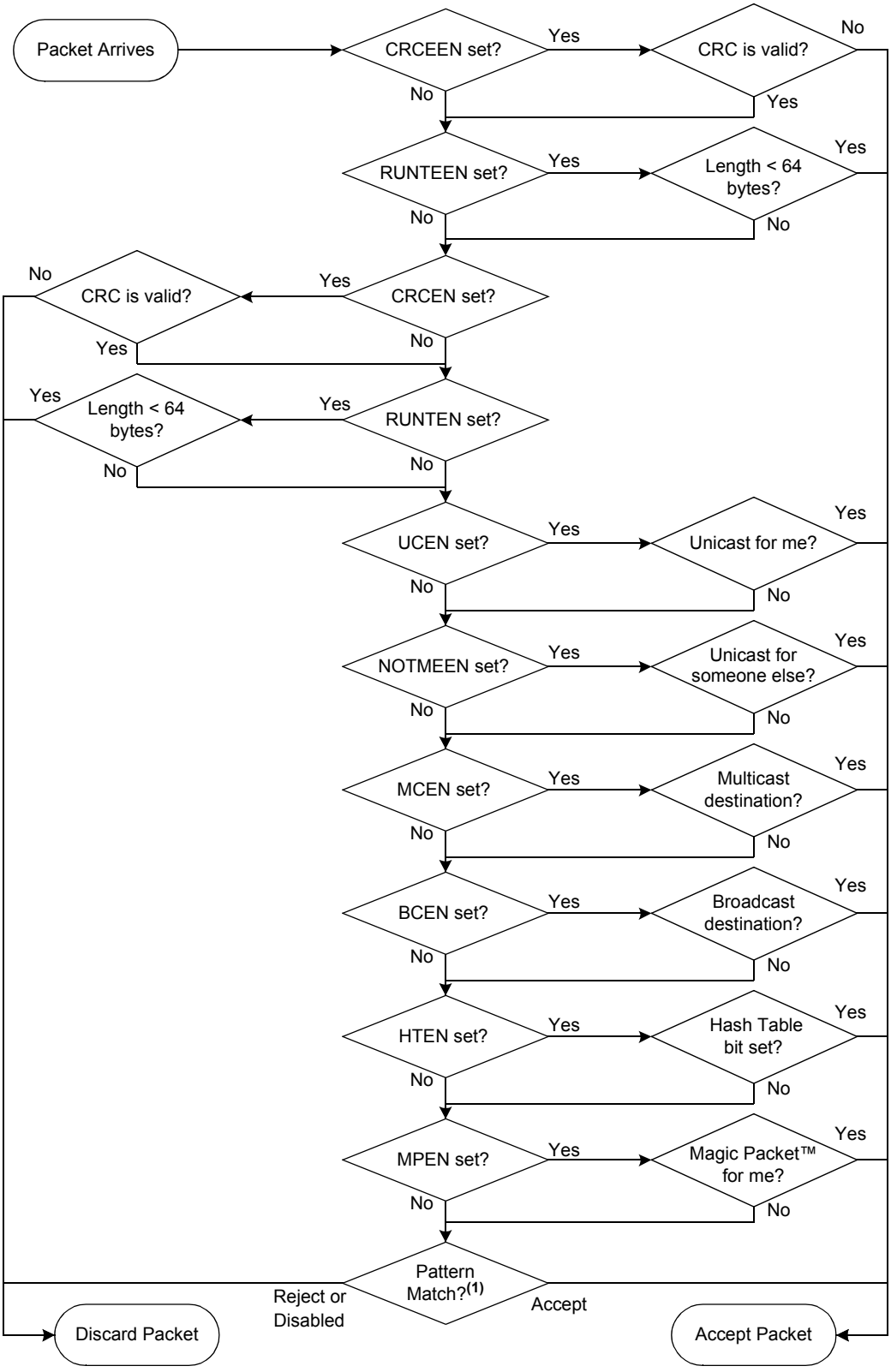
bit 1 **MCEN:** Multicast Destination Collection Filter Enable bit
1 = Accept packets with a Multicast destination address(1)
0 = Filter is disabled

bit 0 **BCEN:** Broadcast Destination Collection Filter Enable bit
1 = Accept packets with a Broadcast destination address of FF-FF-FF-FF-FF-FF(1)
0 = Filter is disabled

Note 1: This filtering decision can be overridden by the CRC Error Rejection filter and Runt Error Rejection filter decisions, if enabled, by CRCEN or RUNTEN.

2: This filtering decision can be overridden by the CRC Error Collection filter and Runt Error Collection filter decisions, if enabled, by CRCEEN or RUNTEEN.

FIGURE 10-1: RECEIVE FILTER DECISION TREE



Note 1: For details on the Pattern Match filter, refer to **Section 10.11 "Pattern Match Collection Filter"**.

10.1 CRC Error Collection Filter

The CRC Error Collection filter allows applications to accept frames with an invalid Frame Check Sequence (FCS). This filter is primarily intended for performing network, cable and layout noise immunity diagnostics.

The filter computes the CRC over incoming frame data and compares the result with the FCS appended at the end of each frame. If the computed CRC does not match the FCS, the filter will accept the frame. If the CRC is correct or the CRC Error Collection filter is disabled, the frame passes through to the next lower priority filter.

This filter is disabled at power-up. To enable the filter, set CRCEEN (ERXFCON<7>). Enabling this filter will cause frames with bit transmission errors and/or invalid data to be accepted into the packet buffer. To comply with IEEE 802.3 specifications, this filter should be disabled.

10.2 Runt Error Collection Filter

The Runt Error Collection filter allows applications to accept frames shorter than 64 bytes (counting from the start of the Ethernet source address to the end of the Frame Check Sequence, inclusive). Runt packets are ordinarily generated by early half-duplex collisions and should not be treated as valid packets. This filter is primarily intended for detecting duplex mismatches or performing network utilization diagnostics.

The filter checks the length of each frame and accepts any frame with a length of 63 or fewer bytes. Frames that are 64 bytes or longer are passed on to the next lower priority filter.

This filter is disabled at power-up. To enable the filter, set RUNTEEN (ERXFCON<5>). If the filter is disabled, all frames will be passed on to the next lower priority filter, regardless of length.

To comply with IEEE 802.3 specifications and properly discard collision fragments, this filter should be disabled.

10.3 CRC Error Rejection Filter

The CRC Error Rejection filter verifies the Frame Check Sequence of incoming frames. If the CRC is invalid, the frame is discarded. Frames with a valid CRC will be passed on to the next filter.

The MAC truncates received frames if they exceed the length specified by the MAC maximum frame length register, MAMXFL. Because the Frame Check Sequence is always transmitted as the last four bytes of any Ethernet frame, reception of an oversize frame that becomes truncated will almost always result in an invalid CRC. When enabled, the CRC Error Rejection filter will discard these truncated frames, as well as collision fragments, and other frames that become corrupted during transmission.

This filter is enabled at power-up. To disable this filter, clear CRCEN (ERXFCON<6>). If the filter is disabled, all frames will be passed on to the next lower priority filter, regardless of CRC validity.

Note:	Enable this filter for normal IEEE 802.3 compliant operation.

10.4 Runt Error Rejection Filter

The Runt Error Rejection filter checks the length of each incoming frame. If the length of the frame is less than the Ethernet minimum of 64 bytes, the frame will be discarded. Frames of 64 bytes or larger will be passed on to the next filter.

This filter is enabled at power-up. To disable this filter, clear RUNTEN (ERXFCON<4>). If the filter is disabled, all frames will be passed on to the next lower priority filter, regardless of length.

Note:	Enable this filter for normal IEEE 802.3 compliant operation.

10.5 Unicast Collection Filter

The Unicast Collection filter checks the destination address of each incoming frame. If the destination address exactly matches the local MAC address, defined by the MAADR registers, the frame will be accepted. If there is a mismatch, the frame will be passed on to the next filter.

This filter is enabled at power-up. To disable this filter, clear UCEN (ERXFCON<3>). If the filter is disabled, all frames will be passed on to the next lower priority filter, regardless of destination address.

10.6 Not-Me Unicast Collection Filter

The Not-Me Unicast Collection filter checks the destination address of incoming frames. If the destination address is a Unicast address, but does not exactly match the contents of the MAADR registers, the frame will be accepted. This will include any frame specifically addressed to another station, but will not include Multicast or Broadcast frames. If the packet is a Multicast, Broadcast or Unicast frame for the local device, the frame will be passed on to the next filter.

This filter is disabled at power-up. To enable this filter, set NOTMEEN (ERXFCON<2>). If the filter is disabled, all frames will be passed on to the next lower priority filter, regardless of destination address.

10.7 Multicast Collection Filter

The Multicast Collection filter checks the destination address of incoming frames. If the Least Significant bit (LSb) of the first byte of the destination address is set, the frame will be accepted. This represents all Multicast frames. If the frame has a Unicast destination, it will be passed on to the next filter.

This filter is disabled at power-up. To enable this filter, set MCEN (ERXFCON<1>). If the filter is disabled, all frames will be passed on to the next lower priority filter, regardless of destination address.

10.8 Broadcast Collection Filter

The Broadcast Collection filter checks the destination address of incoming frames. If the destination address is FF-FF-FF-FF-FF-FF, the frame will be accepted. Frames matching this filter are designated as being broadcast to all nodes that receive them. All frames with other address values will be passed on to the next filter.

This filter is enabled at power-up. To disable this filter, clear BCEN (ERXFCON<0>). If the filter is disabled, all frames will be passed on to the next lower priority filter, regardless of destination address.

10.9 Hash Table Collection Filter

The Hash Table filter accepts frames based on their destination address, and is configurable for up to 64 different hash values. This filter allows the device to accept frames for multiple destination addresses (without accepting all Not-Me traffic as described in **Section 10.6 "Not-Me Unicast Collection Filter"**). It can also be used to accept traffic for one or more specific Multicast groups (without accepting all Multicast traffic as described in **Section 10.7 "Multicast Collection Filter"**). Note that Hash Table collisions are possible, so applications should still verify the MAC address of accepted frames. This filter simply reduces the amount of incoming traffic for these applications.

The filter performs a 32-bit CRC over the six destination address bytes in the packet, using the polynomial, 4C11DB7h. From the resulting 32-bit binary number, a 6-bit value is taken from bits<28:23>. This value, in turn, points to a location in a table formed by the Ethernet Hash Table registers, ETH1 through ETH4. If the bit in that location is set, the packet meets the Hash Table filter criteria and is accepted. The specific pointer values for each bit location in the table are shown in Table 10-1.

An example of the Hash Table operation is shown in Example 10-1. In this case, the destination address, 01-00-00-00-01-2C, produces a Hash Table Pointer value of 34h, which points to bit 4 of ETH4. If this bit is '1', the packet will be accepted. If this Hash Table bit is '0', the packet will be passed to the next lower priority filter.

By extension, if every bit in the Hash Table is set, the filter criteria will always be met, so all packets will be accepted if no higher priority filter has rejected the packet. Similarly, clearing every bit in the Hash Table registers means that the filter criteria will never be met, so all packets will be passed on to the next lower priority filter.

This filter is disabled at power-up. To enable this filter, set HTEN (ERXFCON<15>). If the filter is disabled, all frames will be passed on to the next lower priority filter, regardless of destination address or Hash Table register values.

EXAMPLE 10-1: DERIVING A HASH TABLE LOCATION

Packet Destination Address:

01-00-00-00-01-2C (hex)

Result of CRC-32 with 4C11DB7h:

`1101 1010 0000 1011 0100 0101 0111 0101` (binary)

Pointer Derived from bits<28:23> of CRC Result:

`110100` (binary) or 34 (hex)

Corresponding Hash Table Location:

EHT4<4>

TABLE 10-1: BIT ASSIGNMENTS IN HASH TABLE REGISTERS

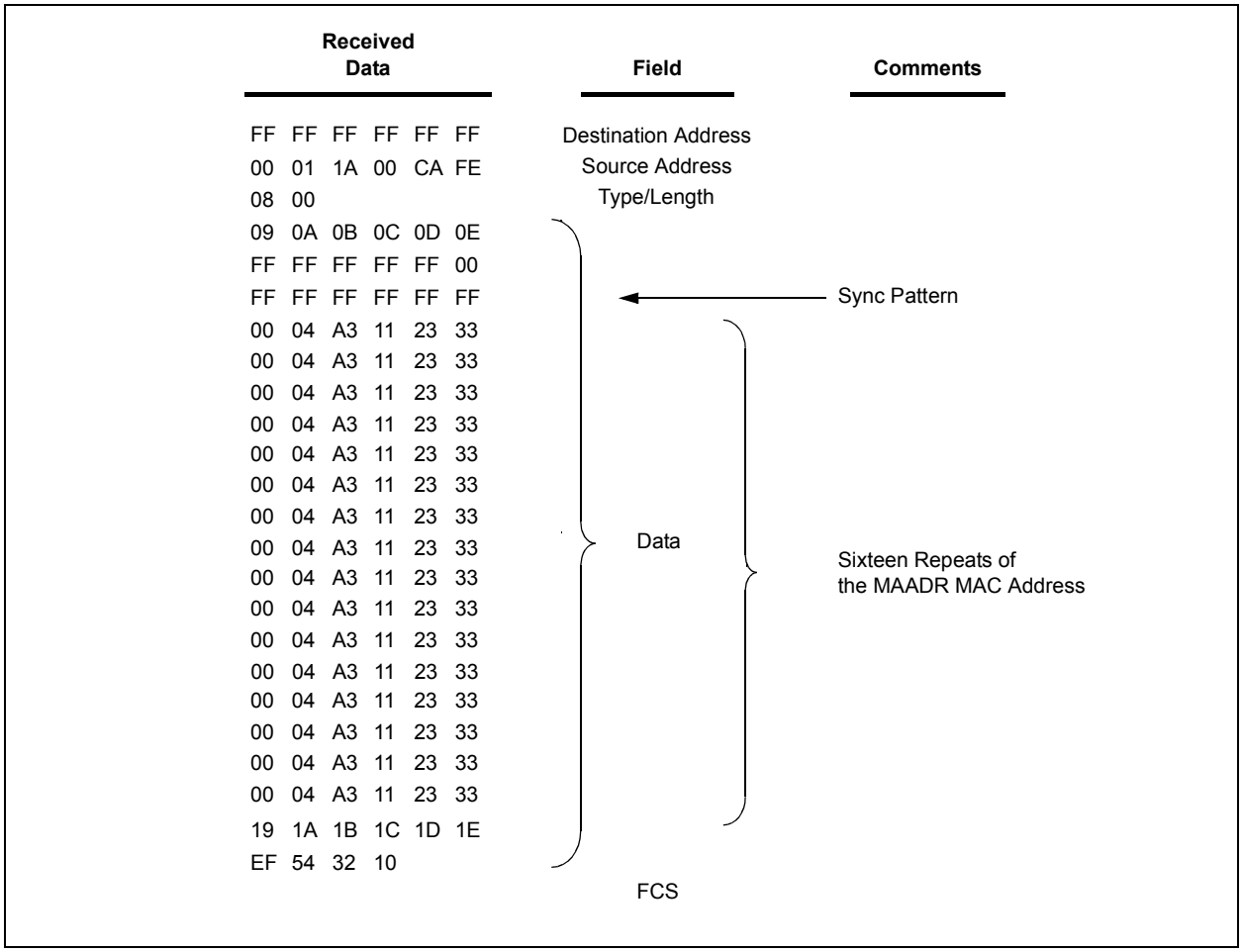
Register	Bit Numbers in Hash Table															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EHT1	0F	0E	0D	0C	0B	0A	09	08	07	06	05	04	03	02	01	00
EHT2	1F	1E	1D	1C	1B	1A	19	18	17	16	15	14	13	12	11	10
EHT3	2F	2E	2D	2C	2B	2A	29	28	27	26	25	24	23	22	21	20
EHT4	3F	3E	3D	3C	3B	3A	39	38	37	36	35	34	33	32	31	30

10.10 Magic Packet™ Collection Filter

The Magic Packet filter scans all packet contents for a Magic Packet pattern for the local MAC address. If a valid Magic Packet pattern is found, then the frame will be accepted. The Magic Packet pattern consists of a sync pattern of six FFh bytes, followed by the MAC address of the station the Magic Packet is intended for, repeated 16 times. See Figure 10-2 for a sample Magic Packet. This pattern may be located anywhere within the packet. Other fields in the packet, such as the destination address or bytes preceding or following the Magic Packet pattern, are ignored.

This filter is disabled at power-up. To enable this filter, set MPEN (ERXFCON<14>). If the filter is disabled or the received packet is not a Magic Packet, the frame will be passed to the next lower priority filter.

FIGURE 10-2: SAMPLE MAGIC PACKET™ FORMAT



10.11 Pattern Match Collection Filter

The Pattern Match filter accepts frames that match or do not match a specific pattern. This filter is useful for accepting frames that contain expected data sequences.

Pattern matching is accomplished by choosing a 64-byte window within the first 128 bytes of a frame, then selecting some or all of those bytes for a checksum calculation. The checksum algorithm is the same as the TCP/IP checksum calculation described in **Section 14.2 "Checksum Calculations"**. This checksum is then compared to the EPMCS register and the result is optionally negated by the NOTPM (ERXFCON<12>) flag.

The Pattern Match filter's control bits, PMEN<3:0> (ERXFCON<11:8>), differ from all other filters in that there are multiple options. The output of the above match can be ANDed with several other conditions. This adds significant flexibility to the filter as it can require both a Pattern Match (or non-match) and other criteria (such as a Broadcast frame or Hash Table match).

To use the Pattern Match filter, the host controller must first program the Pattern Match Offset (EPMO) to select the 64-byte window to be used. Setting this register to 0000h selects the first 64 bytes of the frame, beginning with the first byte of the destination address. Setting 0006h selects byte numbers, 6 through 69, beginning with the first byte of the source address. This window must fall within the first 128 bytes of a frame; the offset value of 1 is not supported, thus, the valid values for EPMO are 0, 2-63.

Note that if the frame length is short enough so that the entire window would not exist in the frame, the filter will automatically fail to match. This is true even if the corresponding mask bits are all '0'.

Then, the host must select the Pattern Match mask bytes by using the EPMM registers. Within this 64-byte window, each byte can be selectively included or excluded from the checksum computation by setting or clearing the respective bit in the Pattern Match mask. A bit set to '1' indicates that the byte is to be included. Data bytes with corresponding mask bits set to '0' are completely removed for the purpose of the checksum calculation (as opposed to treating the data bytes as zero).

Next, write the expected checksum to the EPMCS register. To select frames that match the checksum, clear NOTPM (ERXFCON<12>). To select only frames that fail to match the checksum, set NOTPM to '1'. Finally, set the PMEN bits to '0001b' to require only the Pattern Match criteria, or one of the other values to add additional conditions.

For example, to filter all frames having a particular source MAC address of 00-04-A3-FF-FF-FF:

1. Program the Pattern Match offset to 0000h.
2. Set bits, 6-11, of EPMM1 (assuming all other mask bits are '0').
3. Program the EPMCS register with a checksum value of 5BFCh.
4. Clear NOTPM to require an exact match.

Note that the offset is not programmed to 0006h and the EPMM1<5:0> bits are not set; the checksum would still be 5BFCh. However, in this second case, frames less than 70 bytes would never meet the Pattern Match criteria because there would not be a complete 64-byte window beginning at offset position, 0006h. Another example of a Pattern Match filter is illustrated in Figure 10-3.

The Pattern Match Collection filter is disabled at power-up. Because this filter has the lowest priority of all receive filters, if this filter is disabled or the packet does not meet the configured Pattern Match criteria, the packet is automatically discarded.

10.12 Promiscuous Mode

To accept all incoming frames regardless of content (Promiscuous mode), set the CRCEN, RUNTEN, UCEN, NOTMEEN and MCEN bits. Disable all other filters.

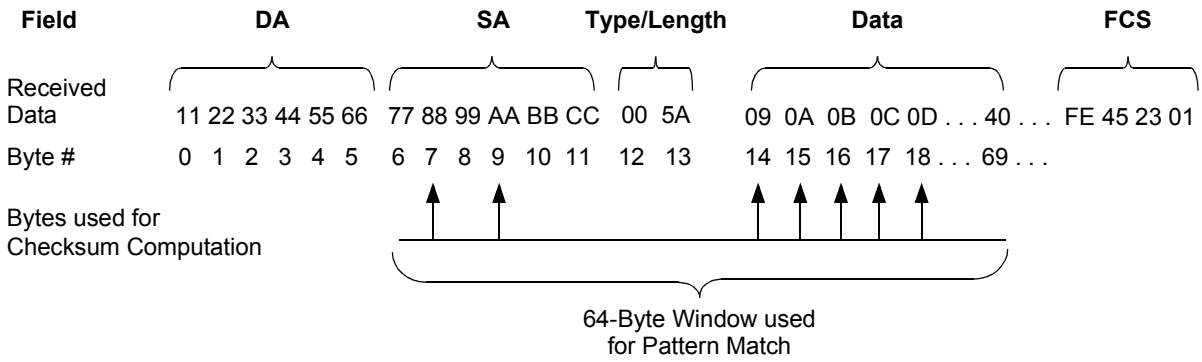
To accept absolutely all recognizable Ethernet frames, including those with errors, set PASSALL (MACCON1<1>) to '1' and set UCEN, NOTMEEN and MCEN in ERXFCON.

In any mode, frames which cannot fit in the receive buffer, or would cause the PKTCNT field (ESTAT<7:0>) to overflow, are still discarded.

FIGURE 10-3: SAMPLE PATTERN MATCH FORMAT

Input Configuration:

EPMO = 0006h
EPMM4:EPMM1 = 0000000000001F0Ah
EPMCS = 563Fh



Values used for Checksum Computation = {88h, AAh, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 00h}
(00h padding byte added by hardware)

Note: Received data is shown in hexadecimal. Byte numbers are shown in decimal format.

NOTES:

11.0 FLOW CONTROL

Flow control provides a mechanism for network stations to temporarily stop transmission of data to themselves. This feature is commonly used to prevent buffer overruns while receiving data.

ENC424J600/624J600 devices are capable of both automatic and manual flow control. The hardware can advertise when it is temporarily unable to receive data and delay transmissions when a remote system does the same. Flow control is supported for both full and half-duplex links. It can either be initiated manually by software, or configured to enable automatically when insufficient space remains in the receive buffer memory.

Flow control operation is configured by the FCOP<1:0> bits (ECON1<7:6>), the AUTOFC bit (ECON2<7>), the RXPAUS and PASSALL bits (MACON1<2:1>), and the EPAUS and ERXWM registers in some modes.

Note:	Flow control is an optional portion of the IEEE 802.3 specification and may not be implemented on all remote devices.

11.1 Modes of Operation

Flow control operation differs between full and half-duplex links. Both modes are supported, but it is important to understand the difference before enabling flow control in an application.

11.1.1 HALF-DUPLEX MODE

When the link is operating in Half-Duplex mode, flow control operates by jamming the network. The node wishing to inhibit transmissions to itself sends a preamble pattern of alternating ones and zeros (55h) on the medium; this is also known as asserting back pressure on the link. Since the link is operating under Half-Duplex mode, all connected nodes must wait before transmitting. If a node does transmit, compliant nodes will detect the collision and wait until the jamming stops to retransmit. This effectively jams the network until flow control is disabled.

If a frame is to be transmitted while flow control is enabled, the ENCX24J600 will stop jamming, wait the standard Inter-Packet Gap (IPG) delay, then attempt to transmit. Because all traffic was previously jammed, several nodes may begin transmitting and several collisions may occur. The hardware will transmit and resume jamming as soon as possible, but it is feasible for other nodes to transmit packets before this happens. This limitation of flow control in half-duplex operation cannot be avoided.

Given the detrimental effect that back pressure based flow control inflicts on a network, along with the possible lack of effectiveness, it is recommended that flow control be avoided in Half-Duplex mode unless the application is used in a closed network environment with proper testing.

When operating in Half-Duplex mode, setting FCOP<1:0> to '00' disables the flow control. Any other combination enables flow control and causes the device to jam the network.

11.1.2 FULL-DUPLEX MODE

Flow control for full-duplex links is much more robust. Instead of jamming the network, a station can send a pause control frame to the remote system. The pause frame is directed to a special Multicast destination address (01-08-C2-00-00-01) and indicates how long the remote node should wait before transmitting again. This time is expressed in units of pause quanta, where one pause quanta is equal to 512 bit times.

While a station is silenced or paused, reception is still enabled. If another pause control frame arrives, any previous value is discarded and the timer restarts using the new pause time value. If the received control frame has a timer value of zero, the pause is terminated and transmission resumes immediately.

When operating in Full-Duplex mode, each combination of FCOP<1:0> has a different effect on transmitting control frames. These combinations are discussed in **Section 11.2.1 "Manual Flow Control"**.

11.1.3 TRANSMITTING AND RECEIVING PAUSE CONTROL FRAMES

The ENCX24J600 automatically processes incoming pause control frames without application intervention. When a pause control frame is received, the MAC internally sets the pause timer. Transmission is inhibited while the timer is active. If an application attempts to transmit a packet during this time, the transmission logic will stall until the time expires (i.e., TXRTS will stay set for longer than normal).

Pause control frames are normally filtered out by the MAC and are not written to the receive buffer. Setting the PASSALL bit (MACON1<1>) alters this behavior and causes these frames to pass through the receive filters. If the frame is accepted, it will be written to the receive buffer. However, setting PASSALL will also cause the MAC to not process the pause control frame. The transmission logic will allow immediate transmission without regard to the remote pause requests.

Before using either automatic or manual flow control, set the pause time value with the EPAUS register. This value controls the pause time value that is transmitted with each pause control frame. Each unit of pause quanta in this register is equal to 512 bit times.

11.2 Manual and Automatic Flow Control

Note:	When flow control is used in conjunction with auto-negotiation, also set the ADPAUS bits (PHANA<11:10>) to '01' during initialization. See **Section 12.0 "Speed/Duplex Configuration and Auto-Negotiation"** for more information.

11.2.1 MANUAL FLOW CONTROL

Manual flow control is enabled by default on device power-up and whenever the AUTOFC bit (ECON2<7>) is cleared. Setting AUTOFC disables manual flow control.

To begin manual flow control in Full-Duplex mode, set FCOP<1:0> to one of the following combinations:

- Idle ('00'): Flow control is disabled or Idle.
- Single Pause ('01'): Transmit one pause frame for the time indicated in EPAUS; automatically returns to the Idle state ('00').
- Continuous Pause ('10'): Periodically transmit pause frames using the value indicated in EPAUS for an indefinite time; must be terminated using End ('11').
- End ('11'): Transmit one pause frame with a timer of 0000h, then return to the Idle state ('00').

In Half-Duplex mode, set FCOP<1:0> to '10' to begin flow control and to '00' to terminate flow control.

11.2.2 AUTOMATIC FLOW CONTROL

When the AUTOFC bit is set, the ENCX24J600 automatically initiates flow control operation when the amount of data in the receive buffer crosses an upper threshold value. Flow control automatically terminates once the amount of pending data shrinks below a lower threshold. These thresholds, or "watermarks", are determined by the upper and lower bytes (respectively) of the Receive Watermark register, ERXWM. These thresholds represent 96-byte blocks.

For example, setting the RXFWM bits to 20h and the RXEWM bits to 10h initiates flow control when more than 3072 bytes of data are present in the buffer. Flow control terminates when fewer than 1536 bytes of data are in the buffer.

In Automatic mode, the value of FCOP<1:0> is controlled by the device and must not be changed by software. These bits, however, can be used as status bits to determine what state the automatic flow control engine is in. A value of '10' indicates flow control is active, while '00' indicates that flow control is Idle.

To use automatic flow control:

1. Set the EPAUS register to indicate how many pause quanta should be specified in each control frame. It is recommended that the default value be used.
2. Set RXFWM<7:0> (ERXWM<15:8>) to indicate when flow control is to begin. When this number of 96-byte blocks is consumed in the receive buffer, the device considers its receive buffer to be full and initiates flow control. Use the default value of 16 if the full threshold is to be set at 1536 bytes.
3. Set RXEWM<7:0> (ERXWM<7:0>) to indicate when flow control is to end. When the number of occupied 96-byte blocks falls below this level, the device considers its receive buffer to be empty enough to receive more data. Use the default value of 15 if the empty threshold is to be set at 1440 bytes.
4. Set AUTOFC (ECON2<7>) to give control of the FCOP<1:0> bits to the automatic flow control hardware.

Note:	Setting RXFWM to be equal to RXEWM (i.e., no hysteresis between full and empty) is not permitted. For automatic flow control to operate correctly, RXEWM must always be at least one less than RXFWM, implying at least 96 bytes of hysteresis.

REGISTER 11-1: MACON1: MAC CONTROL REGISTER 1

R/W-x	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
r	r	—	—	r	r	r	r
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-1
—	—	—	LOOPBK	r	RXPAUS	PASSALL	r
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Reserved:** Write as ‘0’

bit 13-12 **Unimplemented:** Read as ‘0’

bit 11-8 **Reserved:** Write as ‘0’

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4 **LOOPBK:** MAC Loopback Enable bit
1 = Transmitted packets are looped back inside the MAC before reaching the PHY
0 = Normal operation

bit 3 **Reserved:** Write as ‘1’

bit 2 **RXPAUS:** Pause Control Frame Reception Enable bit
1 = Inhibit transmissions when pause control frames are received (normal operation)
0 = Ignore pause control frames which are received

bit 1 **PASSALL:** Pass All Received Frames Enable bit
1 = Control frames received by the MAC are written into the receive buffer if not filtered out
0 = Control frames are discarded after being processed by the MAC (normal operation)

bit 0 **Reserved:** Write as ‘1’

NOTES:

12.0 SPEED/DUPLEX CONFIGURATION AND AUTO-NEGOTIATION

ENC424J600/624J600 devices are capable of operation at 10Base-T and 100Base-TX speeds in Half-Duplex and Full-Duplex modes for each. The speed and Duplex mode can be selected manually, or the part can be configured to automatically select the optimum link parameters based on the capabilities of the link partner. When operating in 10Base-T mode, the part also compensates for incorrect polarity on the TPIN+/- pins (100Base-TX signalling is non-polarized).

In half-duplex operation, only one Ethernet controller may transmit on the physical medium at any given time. If the host controller initiates a transmission while another device is transmitting, the ENCX24J600 will delay until the remote transmitter finishes its packet. Other devices on the medium should do the same while the ENCX24J600 is transmitting. If two controllers begin transmitting at about the same time, a collision will occur. In this case, the data on the medium is corrupt; the ENCX24J600 aborts transmission and attempts to retransmit later.

In full-duplex operation, both nodes may transmit simultaneously, so collisions do not occur. For details about transmitting packets, including collision detection and correction, refer to **Section 9.1 "Transmitting Packets"**.

Speed and Duplex modes are configured in the PHCON1 register (Register 12-1). The PHSTAT1, PHSTAT2 and PHSTAT3 registers (Registers 12-2 through 12-4) provide additional information about the status of the link. The PHANA, PHANLPA and PHANE registers (Registers 12-5 through 12-7) contain information about auto-negotiation status and configuration.

12.1 Manual Configuration

Speed and Duplex modes can be manually selected by disabling auto-negotiation. Manual configuration is enabled by clearing the ANEN bit (PHCON1<12>).

When manual configuration is used, both the Speed and Duplex mode must be selected. Set the SPD100 bit (PHCON1<13>) to select 100Base-TX operation or clear SPD100 to select 10Base-T mode. Set the PFULDPX bit (PHCON1<8>) to configure Full-Duplex mode or clear PFULDPX to use half-duplex operation.

After reconfiguring the Speed and Duplex modes, update the MACON2, MACLCON, MAIPG and MABBIPG registers as described in **Section 8.9 "After Link Establishment"**.

12.2 Auto-Negotiation

Auto-negotiation allows Ethernet devices to agree upon the fastest supported transmission rate. When an Ethernet link is broken, a series of Fast Link Pulses (FLPs) are transmitted periodically to initiate a link. Among other things, these pulses encode information about the node's speed and duplex capabilities.

If a remote partner exists and supports auto-negotiation, it will reply with a burst of FLPs to advertise its own link capabilities. If both devices support 100Base-TX full duplex, the link will be established and that mode will be used. Otherwise, the link falls back to 100Base-TX half duplex, 10Base-T full duplex or 10Base-T half duplex, in that order.

If the remote link partner does not support auto-negotiation, the device will use an algorithm known as Parallel Detection to determine if the link partner is a 10Base-T device or 100Base-TX device. Parallel Detection will optimally resolve the operating speed, however, it will not have any means of learning the duplex state of the remote node. Therefore, the ENCX24J600 PHY will always resort to the half-duplex state when auto-negotiation is not available. A duplex mismatch will occur if the remote device is operating in Full-Duplex mode. To determine whether or not the remote link supports auto-negotiation, check the value of the LPANABL bit (PHANE<0>).

Auto-negotiation is enabled by default at power-up, but can be disabled by clearing the ANEN bit. To restart the auto-negotiation process, set RENEG (PHCON1<9>). After setting RENEG, the hardware automatically clears this bit to '0' immediately.

During auto-negation, the information in the PHANA register is advertised to the link partner by transmitting the information encoded in the Fast Link Pulses. The ANDONE bit (PHSTAT1<5>) is set by the hardware when the auto-negotiation process completes. The value of SPDDPX<2:0> (PHSTAT3<4:2>) indicates which operation mode has been selected. The remote link partner's capabilities are also stored in the PHANLPA register.

Note:	When auto-negotiation is enabled, SPD100 (PHCON1<13>) and PFULDPX (PHCON1<8>) are control only bits. They have no effect on Speed or Duplex modes and do not indicate the current selection when read.

When LINKIF link status change interrupt flag is set, it means auto-negotiation or parallel detection is complete. Once auto-negotiation is complete, the MAC registers related to Duplex mode must be reconfigured. Determine the new Duplex mode by reading the PHYDPX bit (ESTAT<10>). Once this is done, update the MACON2, MACLCON, MAIPG and MABBIPG registers as described in **Section 8.9 "After Link Establishment"**.

REGISTER 12-1: PHCON1: PHY CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
PRST	PLOOPBK	SPD100(1)	ANEN	PSLEEP	r	RENEG	PFULDPX(1)
bit 15							bit 8

R/W-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
r	r	r	r	r	r	r	r
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **PRST:** PHY Reset bit
1 = Perform PHY Reset. Hardware automatically clears this bit to '0' when the Reset is complete.
0 = PHY is not in Reset (normal operation)

bit 14 **PLOOPBK:** PHY Loopback Enable bit
1 = Loopback is enabled
0 = Normal operation

bit 13 **SPD100:** PHY Speed Select Control bit(1)
1 = 100 Mbps
0 = 10 Mbps

bit 12 **ANEN:** PHY Auto-Negotiation Enable bit
1 = Auto-negotiation is enabled. SPD100 and PFULDPX are ignored.
0 = Auto-negotiation is disabled. SPD100 and PFULDPX control the operating speed and duplex.

bit 11 **PSLEEP:** PHY Sleep Enable bit
1 = PHY is powered down
0 = Normal operation

bit 10 **Reserved:** Write as '0', ignore on read

bit 9 **RENEG:** Restart Auto-Negotiation Control bit
1 = Restart the auto-negotiation process. Hardware automatically clears this bit to '0' when the auto-negotiation process starts.
0 = Normal operation

bit 8 **PFULDPX:** PHY Duplex Select Control bit(1)
1 = Full duplex
0 = Half duplex

bit 7 **Reserved:** Write as '0', ignore on read

bit 6-0 **Reserved:** Ignore on read

Note 1: Applicable only when auto-negotiation is disabled (ANEN = 0).

REGISTER 12-2: PHSTAT1: PHY STATUS REGISTER 1

R-0	R-1(1)	R-1(1)	R-1(1)	R-1(1)	R-0	R-0	R-0
r	FULL100	HALF100	FULL10	HALF10	r	r	r
bit 15							bit 8

R-0	R-0	R-0	R/LH-0	R-1(1)	R/LL-0	R-0	R-1(1)
r	r	ANDONE	LRFAULT	ANABLE	LLSTAT	r	EXTREGS
bit 7							bit 0

Legend:	LL = Latch Low bit	U = Unimplemented bit, read as '0'	
R = Readable bit	W = Writable bit	LH = Latch High bit	LL = Latch-Low bit
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Reserved:** Read as '0'

bit 14 **FULL100:** 100Base-TX Full-Duplex Ability Status bit
1 = PHY is capable of 100Base-TX full-duplex operation(1)

bit 13 **HALF100:** 100Base-TX Half-Duplex Ability Status bit
1 = PHY is capable of 100Base-TX half-duplex operation(1)

bit 12 **FULL10:** 10Base-T Full-Duplex Ability Status bit
1 = PHY is capable of 10Base-T full-duplex operation(1)

bit 11 **HALF10:** 10Base-T Half-Duplex Ability Status bit
1 = PHY is capable of 10Base-T half-duplex operation(1)

bit 10-6 **Reserved:** Ignore on read

bit 5 **ANDONE:** Auto-Negotiation Done Status bit
1 = Auto-negotiation is complete
0 = Auto-negotiation is disabled or still in progress

bit 4 **LRFAULT:** Latching Remote Fault Condition Status bit
1 = Remote Fault condition has been detected. This bit latches high and automatically returns to '0' after PHSTAT1 is read.
0 = No remote Fault has been detected since the last read of PHSTAT1

bit 3 **ANABLE:** Auto-Negotiation Ability Status bit
1 = PHY is capable of auto-negotiation(1)

bit 2 **LLSTAT:** Latching Link Status bit
1 = Ethernet link is established and has stayed continuously established since the last read of PHSTAT1
0 = Ethernet link is not established or was not established for a period since the last read of PHSTAT1

bit 1 **Reserved:** Ignore on read

bit 0 **EXTREGS:** Extended Capabilities Registers Present Status bit
1 = PHY has extended capability registers at addresses, 16 thru 31(1)

Note 1: This is the only valid state for this bit; a '0' represents an invalid condition.

REGISTER 12-3: PHSTAT2: PHY STATUS REGISTER 2

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
r	r	r	r	r	r	r	r
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-0	R-1	R-0	R-1	R-1
r	r	r	PLRITY	r	r	r	r
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-5 **Reserved:** Write as '0', ignore on read

bit 4 **PLRITY:** TPIN+/- Polarity Status bit (applies to 10Base-T only)
- 1 = Wiring on the TPIN+/- pins is reversed polarity. PHY internally swaps the TPIN+/- signals to get the correct polarity.
- 0 = Wiring on the TPIN+/- is correct polarity

bit 3-0 **Reserved:** Ignore on read

REGISTER 12-4: PHSTAT3: PHY STATUS REGISTER 3

R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
r	r	r	r	r	r	r	r
bit 15							bit 8

R/W-0	R/W-1	R/W-0	R-0	R-0	R-0	R/W-0	R/W-0
r	r	r	SPDDPX2	SPDDPX1	SPDDPX0	r	r
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-7 **Reserved:** Write as '0', ignore on read

bit 6 **Reserved:** Write as '1'

bit 5 **Reserved:** Write as '0', ignore on read

bit 4-2 **SPDDPX<2:0>:** Current Operating Speed and Duplex Status bits
- 111 = Reserved
- 110 = 100 Mbps, full duplex
- 101 = 10 Mbps, full duplex
- 100 = Reserved
- 011 = Reserved
- 010 = 100 Mbps, half duplex
- 001 = 10 Mbps, half duplex
- 000 = Reserved

bit 1-0 **Reserved:** Write as '0', ignore on read

REGISTER 12-5: PHANA: PHY AUTO-NEGOTIATION ADVERTISEMENT REGISTER

R-0	R-0	R/W-0	R-0	R/W-0	R/W-0	R-0	R/W-1
ADNP	r	ADFAULT	r	ADPAUS1	ADPAUS0	r	AD100FD
bit 15							bit 8

R/W-1	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
AD100	AD10FD	AD10	ADIEEE4	ADIEEE3	ADIEEE2	ADIEEE1	ADIEEE0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **ADNP:** Advertise Next Page Ability bit
`1` = Invalid
`0` = Local PHY does not support auto-negotiation next page abilities

bit 14 **Reserved:** Read as '`0`'

bit 13 **ADFAULT:** Advertise Remote Fault Condition bit
`1` = Local PHY has a Fault condition present
`0` = Local PHY does not have a Fault condition present

bit 12 **Reserved:** Read as '`0`'

bit 11-10 **ADPAUS<1:0>:** Advertise PAUSE Flow Control Ability bits
`11` = Local device supports both symmetric PAUSE and asymmetric PAUSE toward local device
`10` = Local device supports asymmetric PAUSE toward link partner only
`01` = Local device supports symmetric PAUSE only (Normal Flow Control mode)
`00` = Local device does not support PAUSE flow control

bit 9 **Reserved:** Read as '`0`'

bit 8 **AD100FD:** Advertise 100Base-TX Full-Duplex Ability bit
`1` = Local PHY is capable of 100Base-TX full-duplex operation
`0` = Local PHY is incapable of 100Base-TX full-duplex operation

bit 7 **AD100:** Advertise 100Base-TX Half-Duplex Ability bit
`1` = Local PHY is capable of 100Base-TX half-duplex operation
`0` = Local PHY is incapable of 100Base-TX half-duplex operation

bit 6 **AD10FD:** Advertise 10Base-T Full-Duplex Ability bit
`1` = Local PHY is capable of 10Base-T full-duplex operation
`0` = Local PHY is incapable of 10Base-T full-duplex operation

bit 5 **AD10:** Advertise 10Base-T Half-Duplex Ability bit
`1` = Local PHY is capable of 10Base-T half-duplex operation
`0` = Local PHY is incapable of 10Base-T half-duplex operation

bit 4-0 **ADIEEE<4:0>:** Advertise IEEE Standard Selector Field bits
`00001` = IEEE 802.3 Std.
All other values reserved by IEEE. Always specify a selector value of '`00001`' for this device.

REGISTER 12-6: PHANLPA: PHY AUTO-NEGOTIATION LINK PARTNER ABILITY REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
LPNP	LPACK	LPFAULT	r	LPPAUS1	LPPAUS0	LP100T4	LP100FD
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
LP100	LP10FD	LP10	LPIEEE4	LPIEEE3	LPIEEE2	LPIEEE1	LPIEEE0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **LPNP:** Link Partner Next Page Ability bit
1 = Link partner PHY supports auto-negotiation next page abilities
0 = Link partner PHY does not support auto-negotiation next page abilities

bit 14 **LPACK:** Link Partner Acknowledge Local PHY Code Word Status bit
1 = Link partner PHY has successfully received the local PHY abilities saved in PHANA
0 = Link partner PHY has not received the local PHY abilities saved in PHANA

bit 13 **LPFAULT:** Link Partner Remote Fault Condition bit
1 = Link partner PHY has a Fault condition present
0 = Link partner PHY does not have a Fault condition present

bit 12 **Reserved:** Ignore on read

bit 11-10 **LPPAUS<1:0>:** Link Partner PAUSE Flow Control Ability bits
11 = Link partner supports both symmetric PAUSE and asymmetric PAUSE toward local device. Link partner generates and responds to PAUSE control frames. Alternatively, if the local device only supports asymmetric PAUSE, the link partner will respond to PAUSE control frames, but not generate any.
10 = Link partner supports asymmetric PAUSE toward local device only; it can transmit PAUSE control frames, but cannot act upon PAUSE frames sent to it
01 = Link partner supports symmetric PAUSE only, and generates and responds to PAUSE control frames
00 = Link partner does not support PAUSE flow control

bit 9 **LP100T4:** Link Partner 100Base-T4 Ability bit
1 = Link partner PHY is capable of operating in 100Base-T4 mode
0 = Link partner PHY is incapable of operating in 100Base-T4 mode

bit 8 **LP100FD:** Link Partner 100Base-TX Full-Duplex Ability bit
1 = Link partner PHY is capable of 100Base-TX full-duplex operation
0 = Link partner PHY is incapable of 100Base-TX full-duplex operation

bit 7 **LP100:** Link Partner 100Base-TX Half-Duplex Ability bit
1 = Link partner PHY is capable of 100Base-TX half-duplex operation
0 = Link partner PHY is incapable of 100Base-TX half-duplex operation

bit 6 **LP10FD:** Link Partner 10Base-T Full-Duplex Ability bit
1 = Link partner PHY is capable of 10Base-T full-duplex operation
0 = Link partner PHY is incapable of 10Base-T full-duplex operation

bit 5 **LP10:** Link Partner 10Base-T Half-Duplex Ability bit
1 = Link partner PHY is capable of 10Base-T half-duplex operation
0 = Link partner PHY is incapable of 10Base-T half-duplex operation

bit 4-0 **LPIEEE<4:0>:** Link Partner IEEE Standard Selector Field bits
00001 = IEEE 802.3 Std.
All other values are reserved by IEEE. Remote node should also specify this as the selector value.

REGISTER 12-7: PHANE: PHY AUTO-NEGOTIATION EXPANSION REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
r	r	r	r	r	r	r	r
bit 15							bit 8

R-0	R-0	R-0	R/LH-0	R-0	R-0	R/LH-0	R-0
r	r	r	PDFLT	r	r	LPARCD	LPANABL
bit 7							bit 0

Legend:	LH = Latch High bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-5 **Reserved:** Ignore on read

bit 4 **PDFLT:** Parallel Detection Fault Status bit
1 = Parallel detection did not detect a valid link partner; automatically cleared when register is read
0 = Parallel detection is still in progress or a valid link partner is connected

bit 3-2 **Reserved:** Ignore on read

bit 1 **LPARCD:** Link Partner Abilities Received Status bit
1 = PHANLPA register has been written with a new value from the link partner; automatically cleared when register is read
0 = PHANLPA contents have not changed since the last read of PHANE

bit 0 **LPANABL:** Link Partner Auto-Negotiation Able Status bit
1 = Link partner implements auto-negotiation
0 = Link partner does not implement auto-negotiation

NOTES:

13.0 INTERRUPTS

ENC424J600/624J600 devices have multiple interrupt sources tied to a single output pin, allowing the device to signal the occurrence of events to the host controller. The interrupt pin is active-low and is designed for use by host controllers that can detect falling edges. Interrupts can also be used on a polling basis without connecting the interrupt pin. To use interrupts in this manner, monitor the INT bit (ESTAT<15>) on a periodic basis.

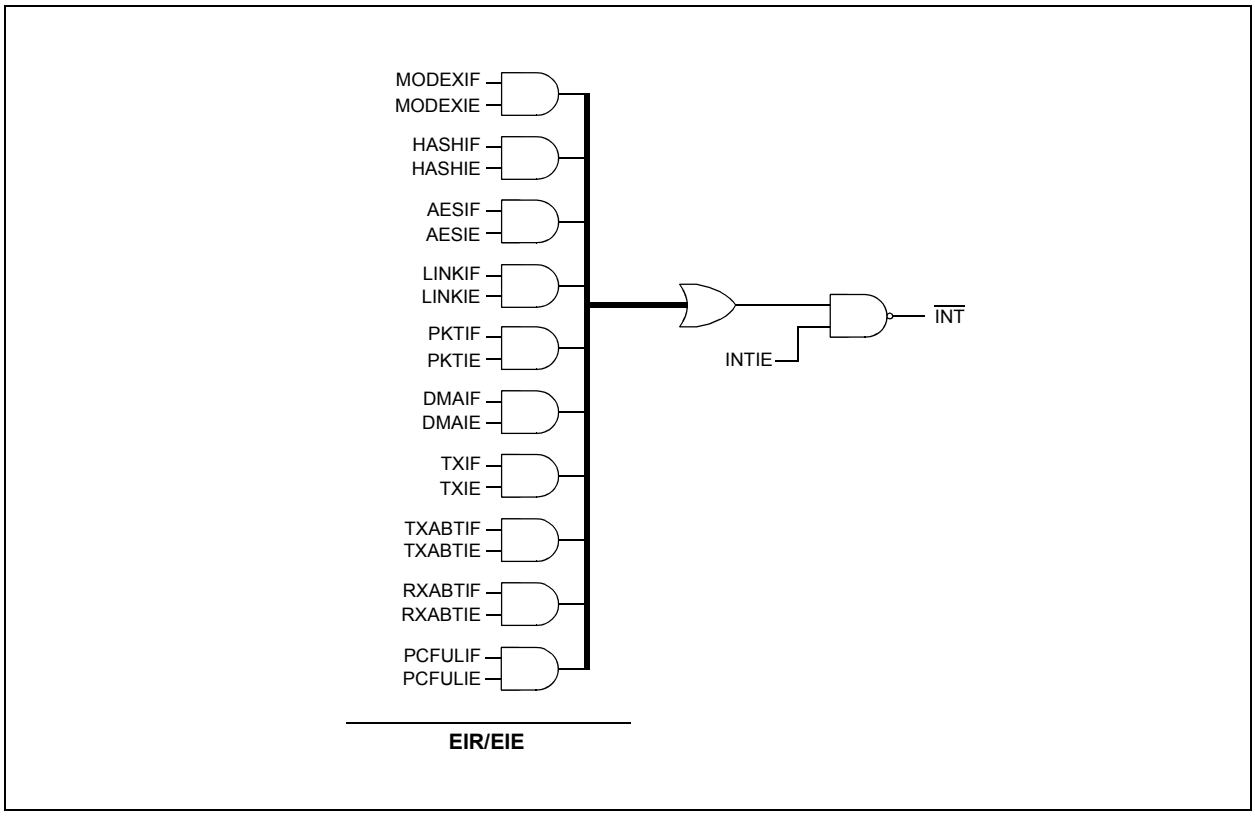
Interrupts are managed by two registers. The EIE register contains the individual interrupt enable bits for each interrupt driven by the MAC and cryptographic components. The EIR register holds the individual interrupt flags. When an interrupt occurs, the corresponding interrupt flag is set. If the interrupt is enabled and the INTIE (EIE<15>) global interrupt enable bit is set, the $\overline{\text{INT}}$ pin is driven low and the INT flag (ESTAT<15>) becomes set. This logic is shown in Figure 13-1.

Even when an interrupt is not enabled, its corresponding interrupt flags are still set when the interrupt condition occurs. This allows the host controller to poll for certain lower priority events while using the interrupt pin for more important tasks.

When an enabled interrupt occurs, the $\overline{\text{INT}}$ pin remains low until all flags causing interrupts are cleared or masked off (the enable bit is cleared). If more than one interrupt source is enabled, the host controller must poll each flag to determine the source(s) of the interrupt. A good practice is for the host controller to clear the Global Interrupt Enable bit, INTIE (EIE<15>), immediately after an interrupt event. This causes the interrupt pin to return to the non-asserted (high) state. Once the interrupt has been serviced, the INTIE bit is set again to re-enable interrupts. If a new interrupt occurs while servicing another, the act of resetting the global enable bit will cause a new falling edge to occur on the interrupt pin and ensure that the host does not miss any events.

When clearing EIR interrupt flags, it is required that bit-oriented operations be used. These include Bit Field Set and Bit Field Clear opcodes for the SPI interface, and using the Bit Set and Bit Clear registers for the PSP interfaces. This procedure ensures that interrupts occurring during the write procedure are not inadvertently missed.

FIGURE 13-1: ENC424J600/624J600 INTERRUPT LOGIC



REGISTER 13-1: EIR: ETHERNET INTERRUPT FLAG REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0
CRYPTEN	MODEXIF	HASHIF	AESIF	LINKIF	r	r	r
bit 15							bit 8

R/W-0	R-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
r	PKTIF	DMAIF	r	TXIF	TXABTIF	RXABTIF	PCFULIF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **CRYPTEN:** Modular Exponentiation and AES Cryptographic Modules Enable bit
- 1 = All cryptographic engine modules are enabled
- 0 = Modular exponentiation and AES modules are disabled and powered down; MD5/SHA-1 hashing is still available

bit 14 **MODEXIF:** Modular Exponentiation Interrupt Flag bit
- 1 = Modular exponentiation calculation is complete
- 0 = No interrupt pending

bit 13 **HASHIF:** MD5/SHA-1 Hash Interrupt Flag bit
- 1 = MD5/SHA-1 hash operation is complete
- 0 = No interrupt pending

bit 12 **AESIF:** AES Encrypt/Decrypt Interrupt Flag bit
- 1 = AES encrypt/decrypt operation is complete
- 0 = No interrupt pending

bit 11 **LINKIF:** PHY Link Status Change Interrupt Flag bit
- 1 = PHY Ethernet link status has changed. Read PHYLNK (ESTAT<8>) to determine the current state.
- 0 = No interrupt pending

bit 10-7 **Reserved:** Ignore on read, don't care on write

bit 6 **PKTIF:** RX Packet Pending Interrupt Flag bit
- 1 = One or more RX packets have been saved and are ready for software processing. The PKTCNT<7:0> (ESTAT<7:0>) bits are non-zero. To clear this flag, decrement the PKTCNT bits to zero by setting PKTDEC (ECON1<8>).
- 0 = No RX packets are pending

bit 5 **DMAIF:** DMA Interrupt Flag bit
- 1 = DMA copy or checksum operation is complete
- 0 = No interrupt pending

bit 4 **Reserved:** Ignore on read, don't care on write

bit 3 **TXIF:** Transmit Done Interrupt Flag bit
- 1 = Packet transmission has completed. TXRTS (ECON1<1>) has been cleared by hardware.
- 0 = No interrupt pending

bit 2 **TXABTIF:** Transmit Abort Interrupt Flag bit
- 1 = Packet transmission has been aborted due to an error. Read the ETXSTAT register to determine the cause. TXRTS (ECON1<1>) has been cleared by hardware.
- 0 = No interrupt pending

REGISTER 13-1: EIR: ETHERNET INTERRUPT FLAG REGISTER (CONTINUED)

bit 1 **RXABTIF:** Receive Abort Interrupt Flag bit
1 = An RX packet was dropped because there is insufficient space in the RX buffer to store the complete packet or the PKTCNT field is saturated at FFh
0 = No interrupt pending

bit 0 **PCFULIF:** Packet Counter Full Interrupt Flag bit
1 = PKTCNT field has reached FFh. Software must decrement the packet counter to prevent the next RX packet from being dropped.
0 = No interrupt pending

REGISTER 13-2: EIE: ETHERNET INTERRUPT ENABLE REGISTER

R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTIE	MODEXIE	HASHIE	AESIE	LINKIE	r	r	r
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-1	R/W-0	R/W-0	R/W-0	R/W-0
r	PKTIE	DMAIE	r(1)	TXIE	TXABTIE	RXABTIE	PCFULIE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **INTIE:** INT Global Interrupt Enable bit
1 = $\overline{INT}$ pin is controlled by the INT status bit (ESTAT<15>)
0 = $\overline{INT}$ pin is driven high

bit 14 **MODEXIE:** Modular Exponentiation Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 13 **HASHIE:** MD5/SHA-1 Hash Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 12 **AESIE:** AES Encrypt/Decrypt Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 11 **LINKIE:** PHY Link Status Change Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 10-7 **Reserved:** Write as '0'

bit 6 **PKTIE:** RX Packet Pending Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 5 **DMAIE:** DMA Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 4 **Reserved:** Ignore on read, don't care on write(1)

bit 3 **TXIE:** Transmit Done Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 2 **TXABTIE:** Transmit Abort Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 1 **RXABTIE:** Receive Abort Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 0 **PCFULIE:** Packet Counter Full Interrupt Enable bit
1 = Enabled
0 = Disabled

Note 1: This bit is read-only and cannot be cleared. Hardware does not modify it.

13.1 Interrupt Sources

ENC424J600/624J600 devices have multiple interrupt sources, each individually selectable. The various interrupt sources are described in the following sections.

For any of the following interrupts to propagate out of the device, the INTIE (EIE<15>) global interrupt enable must be set.

13.1.1 MODULAR EXPONENTIATION COMPLETE

The modular exponentiation complete interrupt occurs when a modular exponentiation operation is completed. This flag is set when MODEXST (ECON1<15>) is cleared. The interrupt should be cleared by software once it has been serviced.

To enable the modular exponentiation complete interrupt, set MODEXIE (EIE<14>).

For more information on the modular exponentiation feature, refer to **Section 15.1 "Modular Exponentiation"**.

13.1.2 MD5/SHA-1 HASH COMPLETE

The MD5/SHA-1 hash complete interrupt occurs when the hashing module completes a block or calculation. The interrupt flag is required when using the hashing engine; therefore, the flag must be cleared by software after the interrupt has been serviced.

To enable the MD5/SHA-1 complete interrupt, set HASHIE (EIE<13>).

For more information on the MD5/SHA-1 hashing feature, refer to **Section 15.2 "MD5 and SHA-1 Hashing"**.

13.1.3 AES COMPLETE

The Advanced Encryption Standard (AES) complete interrupt occurs when a block has been encrypted or decrypted using the AES engine. This flag is set when AESST (ECON1<11>) is cleared. The interrupt should be cleared by software once it has been serviced.

To enable the AES complete interrupt, set AESIE (EIE<12>).

For more information on the Advanced Encryption Standard engine, refer to **Section 15.3 "Advanced Encryption Standard (AES)"**.

13.1.4 LINK CHANGE

The link change interrupt occurs when the PHY link status changes. This flag is set by hardware when a link has either been established or broken between the device and a remote Ethernet partner. The current link status can be read from PHYLNK (ESTAT<8>). The interrupt should be cleared by software once it has been serviced.

To enable the link change interrupt, set LINKIE (EIE<11>).

13.1.5 RECEIVED PACKET PENDING

The received packet pending interrupt occurs when one or more frames have been received and are ready for software processing. This flag is set when the PKTCNT<7:0> (ESTAT<7:0>) bits are non-zero. This interrupt flag is read-only and will automatically clear when the PKTCNT bits are decremented to zero. For more details about receiving and processing incoming frames, refer to **Section 9.0 "Transmitting and Receiving Packets"**.

To enable the received packet pending interrupt, set PKTIE (EIE<6>). The corresponding interrupt flag is PKTIF (EIR<6>).

13.1.6 DMA COMPLETE

The DMA complete interrupt occurs when a DMA operation (either copy or checksum calculation) completes. This flag is set when DMAST (ECON1<5>) is cleared. The interrupt should be cleared by software once it has been serviced.

To enable the DMA complete interrupt, set DMAIE (EIE<5>).

13.1.7 TRANSMIT COMPLETE

The transmit complete interrupt occurs when the transmission of a frame has ended (whether or not it was successful). This flag is set when TXRTS (ECON1<1>) is cleared. The interrupt should be cleared by software once it has been serviced.

To enable the transmit complete interrupt, set TXIE (EIE<3>).

13.1.8 TRANSMIT ABORT

The transmit abort interrupt occurs when the transmission of a frame has been aborted. An abort can occur for any of the following reasons:

- Excessive collisions occurred as defined by the Retransmission Maximum, MAXRET<3:0> bits (MACLCON<3:0>), setting. If this occurs, the COLCNT bits (ETXSTAT<3:0>) will indicate the number of collisions that occurred.
- A late collision occurred after 63 bytes were transmitted. If this occurs, LATECOL (ETXSTAT<10>) will be set.
- The medium was busy and the packet was deferred. If this occurs, EXDEFER (ETXSTAT<8>) will be set.
- The application aborted the transmission by clearing TXRTS (ECON1<1>).

The interrupt should be cleared by software once it has been serviced.

To enable the transmit abort interrupt, set TXABTIE (EIE<2>).

13.1.9 RECEIVE ABORT

The receive abort interrupt occurs when the reception of a frame has been aborted. A frame being received is aborted when the Head Pointer attempts to overrun the Tail Pointer, or when the packet counter has reached FFh. In either case, the receive buffer is full and cannot fit the incoming frame, so the packet has been dropped. This interrupt does not occur when packets are dropped due to the receive filters rejecting a packet. The interrupt should be cleared by software once it has been serviced.

To enable the receive abort interrupt, set RXABTIE (EIE<1>). The corresponding interrupt flag is RXABTIF (EIR<1>).

13.1.10 RECEIVE PACKET COUNTER FULL

The receive packet counter full interrupt occurs when the PKTCNT (ESTAT<7:0>) bits have reached FFh. This indicates that the counter for received frames is full and no more packets can be received. If a packet arrives after this flag is set, it will cause the receive abort interrupt flag to be set. This flag is cleared by hardware once the PKTCNT bits are decremented.

To enable the receive packet counter full interrupt, set PCFULIE (EIE<0>).

13.2 Wake-on-LAN/Remote Wake-up

Wake-on-LAN or remote wake-up is useful for conserving system power. The host controller and other subsystems can be placed in Low-Power mode, then configured to wake-up when a Magic Packet™ is received by the ENC424J600/624J600 devices.

For Wake-on-LAN to operate correctly, the device must not be in Low-Power mode and the receive module must be enabled. When a Magic Packet arrives, the device wakes the host controller via the $\overline{INT}$ signal.

To configure the device for Wake-on-LAN:

1. Set the host controller to wake-up on an external interrupt signal from $\overline{INT}$.
2. Set CRCEN (ERXFCON<6>), RUNTEN (ERXFCON<4>) and MPEN (ERXFCON<14>). Clear all other filter enable bits. This configures the device to only accept Magic Packets.
3. Service all pending packets.
4. Set PKTIE (EIE<6>) and INTIE (EIE<15>) to interrupt when a packet is accepted.
5. Put the host controller and other subsystems to Sleep to save power.

Once a Magic Packet is received, PKTCNT is incremented to '1', causing the device to assert the interrupt signal. When the host wakes up, it needs to restore the normal filter configuration and continue performing its tasks.

For more details about the Magic Packet filter, refer to **Section 10.10 "Magic Packet™ Collection Filter"**.

14.0 DIRECT MEMORY ACCESS (DMA) CONTROLLER

ENC424J600/624J600 devices incorporate a Direct Memory Access (DMA) controller to reduce the burden on the host processor. The module serves the following purposes:

- Copying data from one part of the packet buffer to another.
- Copying data between the packet buffer and one of the memory mapped cryptographic engines.
- Calculating a 16-bit checksum over a block of data, compatible with the checksum used in standard protocols, such as IP and TCP.

In general, the application configures the DMA operation parameters (such as source and destination addresses), then sets the DMAST (ECON1<5>) bit to start the transfer or calculation. The hardware automatically clears the bit when the operation finishes. Additionally, the module can be configured to set an interrupt flag on completion, as detailed in **Section 13.0 "Interrupts"**.

The DMA module follows the same address wrap-around logic as the indirect memory access interfaces described in **Section 3.5.5 "Indirect SRAM Buffer Access"**. When a read or write operation reaches the end of the User-Defined Area (UDA) specified by EUDAND, it automatically wraps to EUDAST. When an operation reaches the end of the general purpose buffer, as indicated by ERXST – 1, it automatically wraps to 0000h. Finally, when an operation reaches the end of the receive buffer (the last address in the physical memory space), it automatically wraps to ERXST. If any of these areas share an ending address, the UDA wrapping will take priority, followed by the general purpose buffer wrapping, then the receive buffer. The wrap-around applies to both the source and destination addresses as an operation progresses.

Although memory is organized by the 16-bit word, the DMA accepts any byte address as the source and destination. It is also capable of operation over an even or an odd number of bytes. Internally, the DMA uses 16-bit accesses, so optimal efficiency is achieved when both the source and destination addresses are even, or when both are odd. Copy operations from an odd source address to an even destination address or vice versa are allowed; however, performance will be diminished by approximately 50% relative to copy operation on addresses that are aligned to each other.

It is recommended that DMA configuration parameters (such as address pointers and operation selection bits) not be modified while DMAST (ECON1<5>) is set and the DMA is active. This requirement is additionally true for the EUDAND Pointer, even if it is not located anywhere within the source or destination memory ranges of the DMA. After processing each memory word, the DMA performs the necessary address wrap-around checks to increment to the next address(es). If the host controller writes to EUDAND at the exact moment the DMA is performing an address wrap-around check, it is possible that the EUDAND register will be in a temporary incoherent state and the DMA source or destination address will wrap to the EUDAST value unintentionally.

Depending upon the operation and the alignment of the source and destination addresses, the DMA module typically requires between one and three clock cycles of OSC1 per 16-bit word. Any DMA operation in progress can be cancelled by clearing the DMAST bit.

14.1 Copying Memory

The DMA can copy any length of data from any address to any destination, including the corner case when the length is zero. Source and destination addresses may be within the implemented SRAM area (0000h through 5FFFh) or the cryptographic data area (7800h through 7C4Fh). It is not possible to use the DMA to read or write from SFRs.

Before initiating the first copy operation, verify that the ETHEN (ECON2<15>) and CLKRDY (ESTAT<12>) bits are set. This does not need to be done for subsequent operations.

To copy data from one location to another:

1. Verify that the values of ERXST, EUDAND and EUDAST (if applicable) are selecting the desired buffer wrapping configuration.
2. Verify that DMAST (ECON1<5>) is clear, indicating that the module is Idle.
3. Set DMACPY (ECON1<4>) to select a copy operation.
4. Optionally, set DMANOCS (ECON1<2>) to prevent a checksum calculation.
5. Set EDMAST to point to the source address.
6. Set EDMADST to point to the destination address.
7. Set EDMALEN to indicate the number of bytes to copy.
8. Set DMAST to initiate the operation.
9. Wait for the hardware to clear DMAST to indicate completion. If the DMA interrupt is enabled, it will be triggered when DMAST is cleared.

Copy operations are performed starting with the first byte or word at the source address and incrementing forward. It is legal to use the DMA to move a block of data backwards in memory, even if the source and destination memory ranges overlap. For example, if a 65-byte packet of data was located starting at memory address 0001h, and the application wished to move the whole packet to address 0000h, it may simply program EDMAST to 0001h and EDMADST to 0000h. The non-overlapping byte at address 00041h will remain unchanged. Similarly, it is also legal to move a block of data from an even aligned address to an immediately prior odd aligned address. For example, moving from 0002h to 0001h will work correctly with the final non-overlapping byte again remaining unchanged.

Since copy operations start at the beginning of the source address range and increment forward (as opposed to starting at the end of the source range and incrementing backwards), it is not possible to move blocks of data forward towards a higher memory address if the source and destination address ranges overlap. To perform this operation, the application must copy the source data to a non-overlapping temporary buffer location and then copy it from the temporary buffer to the final destination.

14.2 Checksum Calculations

The DMA module can compute an IP checksum value over a given range of bytes. Checksums can be calculated over a specific range of memory, or simultaneously as a copy operation progresses. Remember to verify that the values of ERXST, EUDAND and EUDAST (if applicable) are selecting the desired buffer wrapping configuration before starting any checksum operation.

The checksum calculation logic treats the source data as a series of 16-bit big-endian integers. If the source data has an odd number of bytes, a padding byte of 00h will be added for the calculation. (This byte is not copied to the destination if the checksum is part of a copy operation.) The calculated checksum is the 16-bit one's complement of the one's complement sum of all 16-bit integers in the series. For example, if the bytes included in the checksum were {89h, ABh, CDh}, the checksum would begin by calculating 89ABh + CD00h. A carry would occur and the result would be 56ACh. That value would be complemented to yield a checksum of A953h.

To calculate a checksum without copying data:

1. Verify that DMAST (ECON1<5>) is clear, indicating that the module is Idle.
2. Clear DMACPY (ECON1<4>) to prevent a copy operation.
3. Clear DMANOCS (ECON1<2>) to select a checksum calculation.
4. Clear DMACSSD (ECON1<3>) to use the default seed of 0000h. See the paragraph below to seed a checksum with another value.
5. Set EDMAST to point to the source address.
6. Set EDMALEN to indicate the length of the input data.
7. Set DMAST to initiate the operation.
8. Wait for the hardware to clear DMAST to indicate completion. If the DMA interrupt is enabled, it will be triggered when DMAST is cleared.
9. Read the computed checksum from EDMACS.

To calculate a checksum while copying data:

1. Verify that DMAST is clear, indicating that the module is Idle.
2. Set DMACPY to select a copy operation.
3. Clear DMANOCS to select a checksum calculation.
4. Clear DMACSSD to use the default seed of 0000h. See the paragraph below to seed a checksum with another value.
5. Set EDMAST to point to the source address.
6. Set EDMADST to point to the destination address.
7. Set EDMALEN to indicate the number of bytes to copy.
8. Set DMAST to initiate the operation.
9. Wait for the hardware to clear DMAST to indicate completion. If the DMA interrupt is enabled, it will be triggered when DMAST is cleared.
10. Read the computed checksum from EDMACS.

The checksum can be seeded with a previous value if required. Seeding may be useful when attempting to compute a checksum over non-contiguous blocks of data. To seed the checksum, set DMACSSD (ECON1<3>) before initiating the operation. The checksum calculation is seeded with the one's complement of the value contained in EDMACS prior to the start of the DMA operation.

14.3 DMA Performance

The DMA controller can operate at any time, without any regard to other modules in the device. Several factors affect its speed of operation, including:

- Ethernet transmit and receive utilization, especially at 100 Mbps
- SPI or PSP read or write operations to the SRAM
- Operating mode (copy versus checksum only)
- Even-to-odd or odd-to-even source and destination addresses (Copy mode)

Neglecting the time it takes software to program the DMA control SFRs, under typical unloaded conditions, the DMA will have a Checksum Only mode throughput of 100 Mbytes/second. Copy mode (with or without checksum) will achieve a typical throughput of 50 Mbytes/second when the source and destination addresses share the same alignment. Differing source and destination alignment would slow the process to 33.3 Mbytes/second.

Worst case conditions can cut the DMA throughput by no more than half of the typical values.

15.0 CRYPTOGRAPHIC SECURITY ENGINES

To reduce the processing requirements of the host controller, ENC424J600/624J600 devices incorporate three different cryptographic security engines. These security engines perform the types of encryptions, decryptions and mathematical computations that are most commonly used for network security functions. They accelerate the computation of public/private key pair negotiations, message hash authentication and bulk data encryption.

The engines implemented are:

- Modular Exponentiation
- MD5 and SHA-1 Hashing
- Advanced Encryption Standard (AES)

Each engine operates from the cryptography data memory area, shown in **Section 3.4 "Cryptographic Data Memory"**. This memory block is not directly accessible through the SPI or PSP interfaces. Instead, data must be copied into this memory area using the DMA, as described in **Section 14.1 "Copying Memory"**.

The modules are controlled using bits from the ECON2, ECON1 and EIR registers (Registers 8-1, 9-1 and 13-1, respectively). Each of the three modules have separate resources, so they may all be active simultaneously if necessary. Each module also has its own interrupt that can be enabled to signal completion. Refer to **Section 13.0 "Interrupts"** for details on using interrupts.

All of the cryptographic security engines require that the Ethernet enable bit, ETHEN (ECON2<15>) to be set for operation. However, the PHY, which consumes the majority of the device current, can be put to sleep if the application wishes to perform cryptographic operations without Ethernet connectivity. The PHY sleep function is controlled by the PSLEEP bit (PHCON1<11>). For greater information on power down capabilities, refer to **16.0 "Power-Saving Features"**. As a unit, the Modular Exponentiation and AES engines can be disabled to reduce the device's power consumption. This feature is controlled by the CRYPTEN bit (EIR<15>). To enable Modular Exponentiation and AES engines, set CRYPTEN. By default, CRYPTEN is cleared and the modules are disabled on device power-up. The MD5/SHA-1 hashing module remains available regardless of the CRYPTEN state.

15.1 Modular Exponentiation

Modular Exponentiation is the base function for the RSA and Diffie-Hellman algorithms used in public key cryptography. This module computes the value, $Y = X^E \bmod M$, where $0 \le X, Y < M$ and $E > 0$ and $2^{N-1} \le M < 2^N$. N is the chosen operand size of 512, 768 or 1024.

The Modular Exponentiation engine is controlled by the MODEXST bit (ECON1<15>). Setting this bit initiates the calculation. The engine automatically clears the bit when the operation is complete. Clearing the bit in software aborts the calculation in progress and leaves the calculated value in an indeterminate state.

The engine supports operand lengths of 512, 768 or 1024 bits. Operand size is selected with the MODLEN<1:0> bits (ECON2<3:2>). The modulus M and base X can be any value up to the maximum value supported by the chosen operand length. However, if $X \ge M$, the result is not ensured to be correct. Additionally, incorrect results will occur if the exponent E is 0 (this would return the same value as E = 1). Shorter exponents may commonly be used when performing RSA encryption, which typically uses a 17-bit exponent length, or Diffie Hellman computations using a 180-bit exponent.

For the Modular Exponentiation engine to work correctly, the Most Significant (MSb) bit of M must be set. For example, when the MODLEN bits are configured for 1024-bit operation, M must be set such that $2^{1023} \le M < 2^{1024}$. In other words, the modulus must exactly match the chosen operand length of 512 bits, 768 bits or 1024 bits. Although it usually is, the modulus does not have to be a prime number.

All operands must be stored in network (big-endian) byte order. If the base X or exponent E operands are shorter than the selected operand length, they should be right-justified and left-padded with zeros out to the chosen operand length.

To perform a modular exponentiation:

1. Copy the values for X, E and M into the 24-Kbyte SRAM.
2. Set CRYPTEN (EIR<15>) to turn on the Modular Exponentiation module.
3. Use the DMA to transfer E to addresses, 7800h through 783Fh (512-bit), 785F (768-bit) or 787Fh (1024-bit). If the value is shorter than the chosen operand length, left-pad the value with zeros.
4. Use the DMA to transfer the value of X to addresses, 7880h through 78BFh (512-bit), 78DF (768-bit) or 78FFh (1024-bit). If the value is shorter than the chosen operand length, left-pad the value with zeros.
5. Use the DMA to transfer M to addresses, 7900h through 793Fh (512-bit), 795F (768-bit) or 797Fh (1024-bit). The value must not be shorter than the chosen operand length.
6. Set the value of the MODLEN<1:0> (ECON2<3:2>) bits to indicate the size of the operation to be completed.
7. Set MODEXST (ECON1<15> = 1) to initiate the calculation.
8. Wait for the hardware to clear MODEXST to indicate that the operation has completed. The hardware will also set MODEXIF (EIR<14>) and generate an interrupt, if enabled.
9. Use the DMA to transfer the result Y from addresses, 7880h through 78BFh (512-bit), 78DF (768-bit) or 78FFh (1024-bit). Note that this result will be in big-endian format, and if necessary, the result will be left-padded with zeros.
10. If the AES module is not in use, save power by clearing CRYPTEN. Use a Bit Field Clear SPI instruction or write to the EIRCLR register to clear this bit without corrupting any interrupt flags.

Assuming the operating length remains constant, the exponent E and modulus M are retained within the Modular Exponentiation engine and can be reused for future operations without being reloaded.

15.1.1 MODULAR EXPONENTIATION PERFORMANCE

The time required to compute the Modular Exponentiation result depends on three factors:

- the “active” length of the exponent E.
- the density of ‘1’ bits in the exponent E.
- the length of the operands

The time required for typical operations is summarized in Table 15-1.

TABLE 15-1: TYPICAL MODULAR EXPONENTIATION PERFORMANCE

Usage	Operands *(M/E)*	Time
DH	768/180	50.2 ms
DH	1024/180	89.0 ms
RSA	512/512	63.7 ms
RSA	768/768	214.0 ms
RSA	1024/1024	506.2 ms

15.2 MD5 and SHA-1 Hashing

The MD5 and SHA-1 hash engines implement one-way message digest functions. These functions take an unlimited amount of data and produce a digest of either 128 or 160 bits (for MD5 and SHA-1, respectively). They are frequently used for verification and integrity purposes.

Both hashing engines share the same resources, so only one operation may be active at a time. The current operation is selected by the SHA1MD5 (ECON2<12>) bit. This bit should be configured before using the module. Context switching is supported by the engine for applications that require the capability to switch between two or more hashing operations.

The HASHOP (ECON1<13>) bit configures the initialization values. When starting a new hash calculation, clear this bit to reset the initialization values. Using this bit to load a previously saved context is described in **Section 15.2.3 “Context Switching”**. The value of the HASHOP bit may not be changed once the HASHEN bit is set, so it must be configured first.

Set the HASHEN (ECON1<14>) bit to enable the module and begin transferring data. Once this bit is set, all data copied to the module through the DMA will be added to the hash calculation. Data should be written beginning at the Hash Data In address, 7A00h. After copying 64 bytes, the application must pause and wait for the HASHIF (EIR<13>) bit to be set by the hardware. This flag indicates that the hardware has completed processing for that block. The application should then clear the HASHIF flag and continue with the next block, beginning transfers again at address 7A00h.

Before the final DMA transfer is started, the application must set the HASHLST bit (ECON1<12>). When this bit is set, and a DMA transfer is initiated to the hash engine, the engine pads the input appropriately for the selected algorithm and calculates the final result. Once the HASHIF flag is set, the message digest is available in Digest/State Out, beginning at address 7A70h.

The application must wait for the HASHIF flag after every 64-byte block, but all 64 bytes need not be transferred in one operation. For example, it is possible to transfer 16 bytes in one operation and the remaining 48 in a second. However, it is required that the DMA stop copying data once a full 64-byte block is created. For example, if 16 bytes are transferred in one operation, and 52 in the next (for a total of 68 bytes), then the final four bytes will be lost and the output will be incorrect.

Note that the Hash Data In memory is not physically implemented, nor is it accessible for reading. Transfers to any address in the range of 7A00h to 7A3Fh instruct the DMA to write directly to the hash engine. Therefore, if 32 bytes are copied, beginning at 7A00h, a subsequent write of 32 bytes to the same address will not overwrite the previously written data. Instead, the two 32-byte writes are appended to form a single 64-byte block and the hashing process begins. When making multiple transfers as part of a single 64-byte block, the second and subsequent transfers may begin, either at their sequential location, or they may all use the same destination address of 7A00h.

With the exception of the final transfer, all data transfers to the hash engine must be of an integral length of 4 bytes. For example, chunks of 4, 8, 12, 16, etc. are legal, while DMA transfers of length 1, 2, 3, 5, 6, 7, 9, 10, 11, etc. are illegal. Optimal DMA copy performance is also achieved when the source address is word or even aligned. To allow for hashes to be computed over any length of data, the integral length of 4 restriction does not apply to the last transfer (when HASHLST is set).

15.2.1 MD5 HASHING

The module implements the MD5 function, as described in the Internet Engineering Task Force RFC 1321, *"The MD5 Message Digest Algorithm"*. The resulting digest is 128 bits (16 bytes) in length and is left-justified in the result space.

To calculate an MD5 digest:

1. Clear SHA1MD5 (ECON2<12>), HASHOP (ECON1<13>) and HASHLST (ECON1<12>).
2. Set HASHEN (ECON1<14>).
3. Clear HASHIF (EIR<13>).
4. Use the DMA to transfer exactly 64 bytes to address 7A00h. This transfer may be split into multiple transactions if each copy operation is an integral length of 4 and the net of all transfers is 64 bytes.
5. Wait for HASHIF to be set.
6. Repeat steps 3 through 5 until fewer than 64 bytes remain.
7. Clear HASHIF.
8. Set HASHLST (ECON1<12>).
9. Use the DMA to transfer the remaining bytes to address 7A00h.
10. Wait for HASHIF to be set.
11. Use the DMA to transfer the resulting 16-byte hash from address 7A70h. This 128-bit hash will be in big-endian byte order.
12. Clear HASHEN.

Step 5 will take 500 ns from the time the DMA completes the transfer. Under worst case conditions, the DMA will take 3.94 μs to copy a block of 64 bytes after the DMAST bit is set. Therefore, for maximum performance, applications may choose to omit step 3 and replace step 5 with a processor enforced wait of at least 4.5 μs between the start of a DMA copy operation and the start of the next DMA copy operation of 64 bytes.

Steps 7 and 10 may also be optimized. However, the wait period should be extended to no less than 5.6 μs as the hardware requires extra time to perform an extra padding step as required by the MD5 algorithm.

Step 9 can be split into multiple DMA copy transactions if step 8 is held off until immediately before the very last DMA copy operation is performed.

15.2.2 SHA-1 HASHING

The module implements the SHA-1 function as described in the NIST Federal Information Processing Standard (FIPS) Publication 180-1. The resulting digest is 160 bits (20 bytes) in length.

To calculate a SHA-1 digest:

1. Set SHA1MD5 (ECON2<12>). Clear HASHOP and HASHLST (ECON1<13:12>).
2. Set HASHEN (ECON1<14>).
3. Clear HASHIF (EIR<13>).
4. Use the DMA to transfer exactly 64 bytes to address 7A00h. This transfer may be split into multiple transactions if each copy operation is an integral length of 4 and the net of all transfers is 64 bytes.
5. Wait for HASHIF to be set.
6. Repeat steps 3 through 5 until fewer than 64 bytes remain.
7. Clear HASHIF.
8. Set HASHLST (ECON1<12>).
9. Use the DMA to transfer the remaining bytes to address 7A00h.
10. Wait for HASHIF to be set.
11. Use the DMA to transfer the resulting 20-byte hash from address 7A70h. This 160-bit hash will be in big-endian byte order.
12. Clear HASHEN.

Like the MD5 hashing case, steps 3, 5, 7 and 10 can be optimized by replacing them with enforced wait periods. However, SHA-1 is slightly slower than MD5 so a wait period of at least 4.7 μs should be used in place of step 5 and at least 5.8 μs for step 10.

Also, like the MD5 case, step 9 can be split into multiple DMA copy transactions if step 8 is held off until immediately before the very last DMA copy operation is performed.

15.2.3 CONTEXT SWITCHING

At each 64-byte boundary, the current output state can be read from the module. This output state can be stored in memory elsewhere, then loaded back into the module at a later time to continue the hash. Using this feature allows the engine to alternate between calculating two or more digests simultaneously.

To make use of the context switching capability:

1. Initiate a hash calculation.
2. After hashing an integral number of 64-byte blocks, wait for the HASHIF flag to be set.
3. Read the current context from the module and store it elsewhere in memory. The context includes the Digest/State Out and Length State Out values, which comprise the 28 bytes starting at address 7A70h.
4. Once the context has been saved, clear the HASHEN (ECON1<14>) bit to disable the module. The module is now available to be used by other operations.

When the application is ready to resume the previous calculation, restore the context to the Initialization Vector/State In and Length State In values, beginning at 7A40h. Then, set the HASHOP (ECON1<13>) bit to indicate that a previous state is to be loaded from memory rather than initializing a new calculation. Once this bit is set, setting the HASHEN bit allows the hash operation to proceed as usual.

After the context has been saved, the module may be used for a different type of hash (MD5 instead of SHA-1, or vice versa). When loading a context back into the module, verify that SHA1MD5 (ECON2<12>) selects the correct hash operation.

To switch the context during a calculation:

1. Configure SHA1MD5 (ECON2<12>) to select the correct operation.
2. Clear HASHOP (ECON1<13>) to begin a new hash.
3. Set HASHEN (ECON1<14>).
4. Clear HASHIF (EIR<13>).
5. Use the DMA to transfer exactly 64 bytes to address 7A00h. This transfer may be split into multiple transactions if each copy operation is an integral length of 4 and the net of all transfers is 64 bytes.
6. Wait for HASHIF to be set.
7. Repeat steps 4 through 6 for as many complete 64-byte blocks as are ready to be hashed.
8. Use the DMA to transfer the resulting 28 bytes of context data, beginning at address 7A70h, to another location in memory.
9. Clear HASHEN.
10. Use the module for other operations as necessary.
11. Configure SHA1MD5 as in step 1.
12. Use the DMA to transfer the 28 bytes of stored context to address 7A40h.
13. Set HASHOP and HASHEN to resume a previous calculation.
14. Continue using the module as previously described by hashing more data, then either saving the state or completing the calculation.

It is important to note that the Digest/State Out only contains either a Digest or a State Out initialization vector, but not both. If the HASHLST bit is set before the final DMA transfer, the value will indicate the final digest of all data processed so far. This digest is not a valid initialization vector and cannot be used to resume the hash. This is true even if the final transfer filled the buffer to a 64-byte boundary. Likewise, if HASHLST is clear before the final DMA transfer, the value can only be used as an initialization vector. It will not be a valid hash of the message so far. Therefore, applications that require the capability to calculate a hash, add more data and continue, should buffer up to 64 bytes in memory. Only perform the hash operation on a block once the 65th byte is ready to be hashed. This allows the application to select whether a Digest or a State Out initialization vector is desired before hashing a block. Provided the context is stored, the application could request a digest, then reload the context and retransfer the data (beginning at the most recent 64-byte boundary) to continue the hashing operation where it was last stopped.

15.2.4 MD5/SHA-1 HASH PERFORMANCE

The implications noted in **Section 15.2.1 “MD5 Hashing”** and **Section 15.2.2 “SHA-1 Hashing”** are that the hashing engine is extremely fast and net throughput is primarily limited by the DMA. Using an open-loop method of skipping DMA and hash status checking, it is possible to attain a net hashing throughput of 13.6 Mbytes/second (108 Mbits/second). Practical considerations, such as the time it takes to send and receive the data between the Ethernet and host microcontroller, will generally play a bigger roll in the total application performance.

15.3 Advanced Encryption Standard (AES)

The AES engine implements the Advanced Encryption Standard (originally known as Rijndael), as described in the NIST Federal Information Processing Standard Publication 197. This module can be used to encrypt or decrypt data using a known secret key. Context switching is supported for applications that require the capability to alternate between two or more operations or keys.

AES is a block cipher that must operate over 128-bit (16-byte) blocks. The application must apply any necessary padding, or strip any extraneous output bytes, as dictated by the desired padding scheme. No support for padding is included in the engine.

15.3.1 KEY SUPPORT

The AES engine supports 128, 192 and 256-bit key sizes. Keys for AES are symmetric, meaning both parties must agree on a shared secret before the algorithm can be used. This is typically accomplished using an asymmetric algorithm, such as RSA, and/or is handled by a higher level protocol, such as Secure Socket Layer (SSL) or Transport Layer Security (TLS).

To load an encryption key:

1. Verify that AESST (ECON1<11>) is clear, indicating that the engine is Idle.
2. Configure AESLEN<1:0> (ECON2<1:0>) to select the correct key size.
3. Use the DMA to transfer the key data to address 7C00h. Keys shorter than 256 bits should be left-aligned.

AES generates a series of roundkeys from the encryption key using an expansion function. While encryption begins at the first of these keys, decryption must start from the last one. The AES module includes a key expander, which calculates the roundkeys as needed by the encryption engine. To calculate the last roundkey before beginning decryption, the engine must first be operated in Encryption mode for one block.

To initialize decryption using a known encryption key:

1. Verify that AESST is clear, indicating that the engine is Idle.
2. Configure AESLEN<1:0> to select the correct key size.
3. Use the DMA to transfer the encryption key to address 7C00h. Keys shorter than 256 bits should be left-aligned.
4. Configure AESOP<1:0> (ECON1<10:9>) to '00'.
5. Set AESST to initiate the key expansion.
6. Wait until the hardware clears the AESST flag.

15.3.2 CONTEXT SWITCHING

After each block is complete, the internal state may be saved in order to switch encryption keys or operations. Context switching may only be performed when AESST (ECON2<11>) is clear, indicating that the engine is Idle. The values comprising the context varies depending on the mode selected. The sections describing each mode details which values must be saved.

15.3.3 BLOCK MODES

Block ciphers are commonly used in one of five modes as described by the NIST Special Publication 800-38A, "*Recommendations for Block Cipher Modes of Operation: Methods and Techniques"*. The use of the AES engine in each of these modes is described in the following sections.

Four modes are natively supported in hardware:

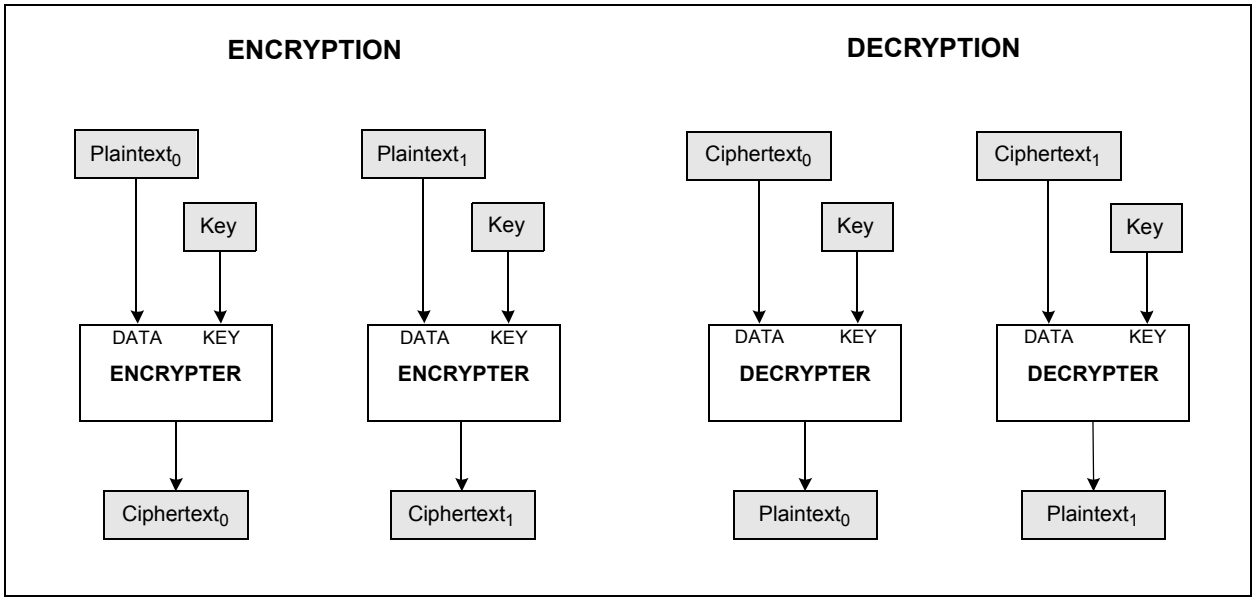
- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC)
- Cipher Feedback (CFB)
- Output Feedback (OFB)

The fifth mode, Counter (CTR), can be used with the addition of software support for the counter.

15.3.3.1 Electronic Code Book Mode (ECB)

The Electronic Code Book mode applies the AES encryption function directly to each plaintext block. No feedback is included, so the encryption of each block is completely independent of any previous block. Assuming a given session key, any plaintext block will always yield to the same ciphertext block (and vice versa). If this is an undesirable property, a different block mode should be selected. Figure 15-1 shows the use of ECB mode for encryption and decryption.

FIGURE 15-1: ECB ENCRYPTION AND DECRYPTION



To encrypt a block using ECB mode:

1. Load the encryption key as described in **Section 15.3.1 "Key Support"**.
2. Set AESOP<1:0> (ECON1<10:9>) to '00'.
3. Copy the plaintext message to TEXTA at 7C20h.
4. Set AESST (ECON1<11>) to initiate the encryption.
5. Wait for the hardware to clear AESST.
6. Read the ciphertext message from TEXTA at 7C20h.
7. Repeat steps 3 through 6 for subsequent blocks.

To decrypt a block using ECB mode:

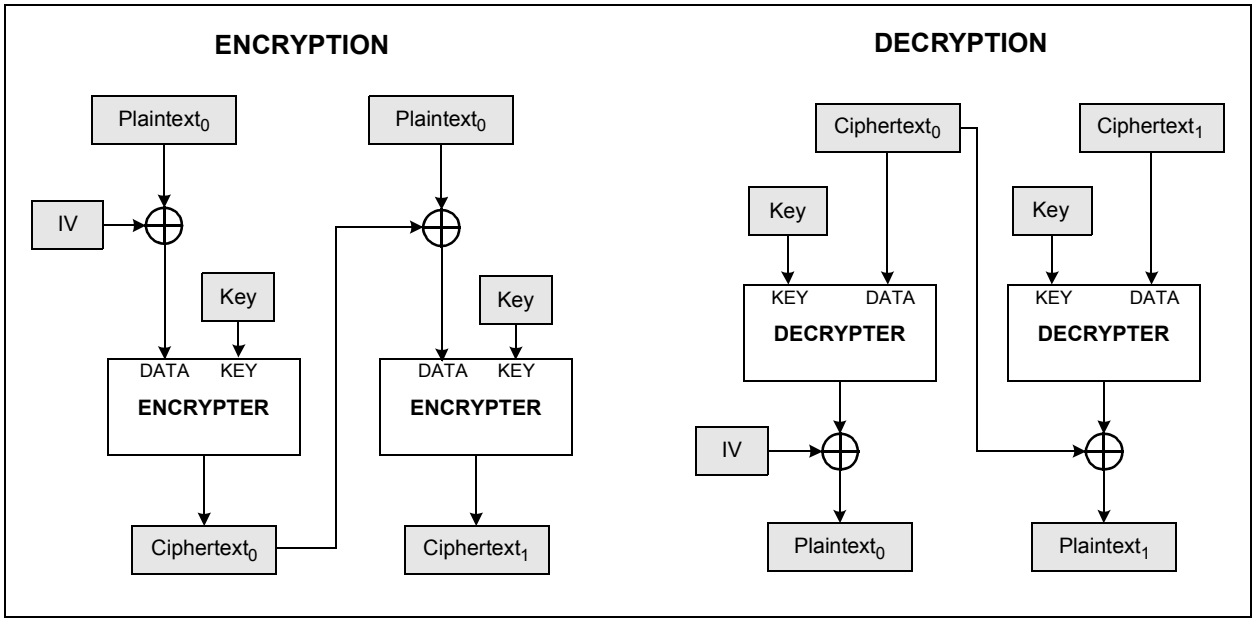
1. Initialize the decryption key as described in **Section 15.3.1 "Key Support"**.
2. Set AESOP<1:0> to '10'.
3. Copy the ciphertext message to TEXTA at 7C20h.
4. Set AESST to initiate the decryption.
5. Wait for the hardware to clear AESST.
6. Read the plaintext message from TEXTA at 7C20h.
7. Repeat steps 3 through 6 for subsequent blocks.

The context for ECB mode includes only the encryption key. No additional context data needs to be saved.

15.3.3.2 Cipher Block Chaining Mode (CBC)

The Cipher Block Chaining mode uses feedback from the encryption output to further obscure the ciphertext data. During encryption, the first block uses an Initialization Vector (IV) which is XORed with the plaintext data. The output of this XOR function is then encrypted using the AES key and this ciphertext becomes the IV for the next block. Under CBC mode, each subsequent block depends on the previous block. Therefore, identical subsequent plaintext blocks use a different IV, and therefore, yield different ciphertext blocks. Figure 15-2 shows the use of CBC mode for encryption and decryption.

FIGURE 15-2: CBC ENCRYPTION AND DECRYPTION



To encrypt a block using CBC mode:

1. Load the encryption key as described in **Section 15.3.1 "Key Support"**.
2. Set AESOP<1:0> (ECON1<10:9>) to '01'.
3. Copy the Initialization Value (IV) to TEXTA at 7C20h.
4. Copy the plaintext message to TEXTB at 7C30h.
5. Set AESST (ECON1<11>) to initiate the encryption.
6. Wait for the hardware to clear AESST.
7. Read the ciphertext message from TEXTA at 7C20h.
8. Repeat steps 4-7 for subsequent blocks. The ciphertext from the previous block automatically becomes the IV for the following block.

To decrypt a block using CBC mode:

1. Initialize the decryption key as described in **Section 15.3.1 "Key Support"**.
2. Set AESOP<1:0> to '10'.
3. Copy the Initialization Value (IV) to TEXTB at 7C30h.
4. Copy the ciphertext message to TEXTA at 7C20h.
5. Set AESST to initiate the decyption.
6. Wait for the hardware to clear AESST.
7. Read the plaintext message from XOROUT at 7C40h.
8. If another block is to be decrypted, copy the ciphertext message from this block to TEXTB at 7C30h. The ciphertext from this block becomes the IV for the following one.
9. Repeat steps 4-8 for subsequent blocks.

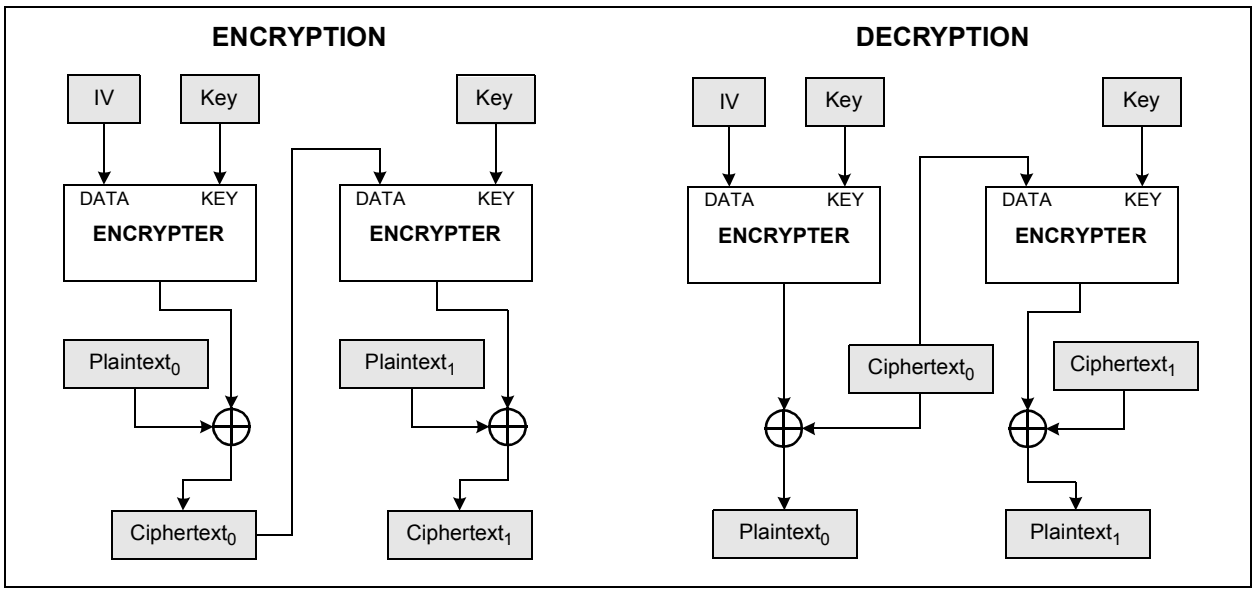
The context for CBC mode consists of both the AES encryption key and the ciphertext from the most recent block. Save the ciphertext from the previous block to be used as the IV when resuming the operation for additional blocks.

15.3.3.3 Cipher Feedback Mode (CFB)

Cipher Feedback mode is similar to CBC in that the ciphertext becomes the Initialization Vector (IV) for the subsequent block. However, in CFB mode, only the IV is encrypted, then XORed with the plaintext to form the ciphertext. For the second and subsequent blocks, the ciphertext is passed through the encryption function again, then XORed with the next plaintext block to become the ciphertext. Like CBC mode, each subsequent block depends on the previous block. Therefore, identical subsequent plaintext blocks will use a different IV, and therefore, yield different ciphertext blocks. Figure 15-3 shows the use of CFB mode for encryption and decryption.

Note:	Only 128-bit CFB mode is natively supported. Other less common implementations, including 1-bit and 8-bit CFB modes, could be accomplished with support software, but are not detailed here.

FIGURE 15-3: CFB ENCRYPTION AND DECRYPTION



To encrypt a block using CFB mode:

1. Load the encryption key as described in **Section 15.3.1 "Key Support"**.
2. Set AESOP<1:0> (ECON1<10:9>) to '00'.
3. Copy the Initialization Value (IV) to TEXTA at 7C20h.
4. Set AESST (ECON1<11>) to initiate the encryption.
5. Copy the plaintext message to TEXTB at 7C30h.
6. Wait for the hardware to clear AESST.
7. Read the ciphertext message from XOROUT at 7C40h.
8. If more blocks need to be encrypted, set AESOP<1:0> to '01'. This causes the engine to read from XOROUT rather than TEXTA.
9. Set AESST to initiate the encryption.
10. Copy the plaintext message to TEXTB at 7C30h.
11. Wait for the hardware to clear AESST.
12. Read the ciphertext message from XOROUT at 7C40h.
13. Repeat steps 9 through 12 for subsequent blocks. The ciphertext from the previous block automatically becomes the IV for the following block.

To decrypt a single block using CFB mode:

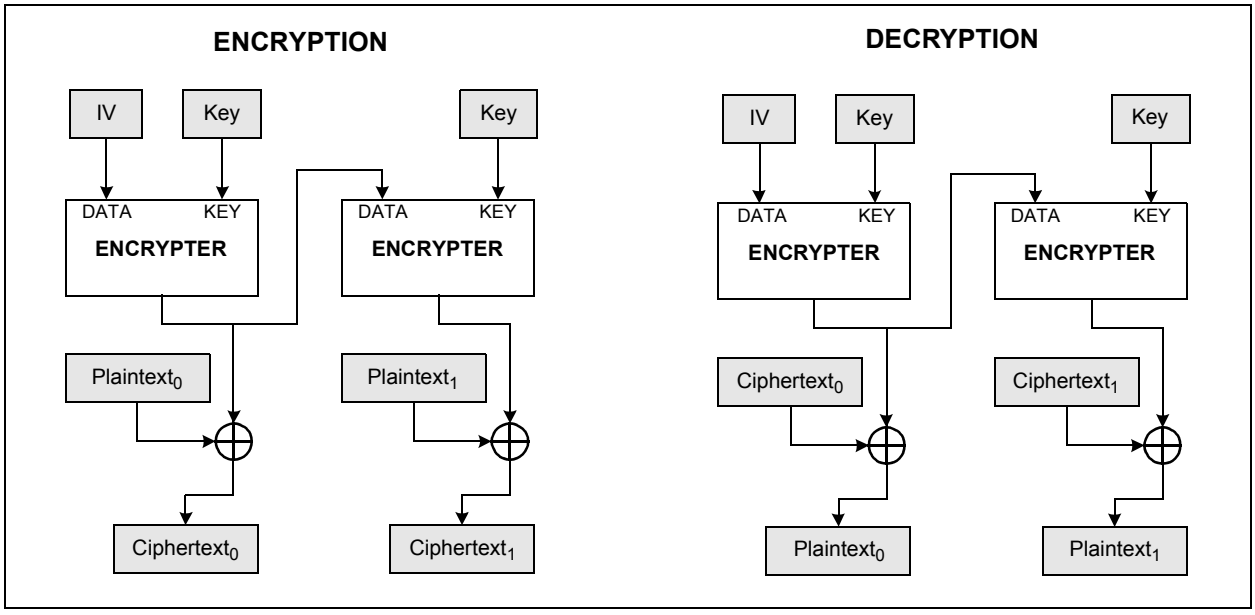
1. Load the encryption key as described in **Section 15.3.1 "Key Support"**.
2. Set AESOP<1:0> to '00'.
3. Copy the Initialization Value (IV) to TEXTA at 7C20h.
4. Set AESST to initiate the encryption.
5. Copy the ciphertext message to TEXTB at 7C30h.
6. Wait for the hardware to clear AESST.
7. Read the plaintext message from XOROUT at 7C40h.
8. To decipher additional blocks, copy the previous block's ciphertext to TEXTA, then repeat steps 4 through 7.

The context for CFB mode consists of both the AES encryption key and the ciphertext from the most recent block. Save the ciphertext from the previous block to be used as the IV when resuming the operation for additional blocks.

15.3.3.4 Output Feedback Mode (OFB)

Output Feedback mode is nearly identical to CFB mode, except that in OFB mode, the Initialization Value (IV) for subsequent blocks is the output of the AES operation, not the ciphertext. The IV is encrypted using the AES engine, then XORed with the plaintext to form the ciphertext. Like CBC and CFB modes, identical subsequent plaintext blocks will use a different IV, and therefore, yield different ciphertext blocks. However, unlike CBC and CFB modes, this IV does not depend on the plaintext. Figure 15-4 depicts the use of OFB mode for encryption and decryption.

FIGURE 15-4: OFB ENCRYPTION AND DECRYPTION



To encrypt a block using OFB mode:

1. Load the encryption key as described in **Section 15.3.1 "Key Support"**.
2. Set AESOP<1:0> (ECON1<10:9>) to '00'.
3. Copy the Initialization Value (IV) to TEXTA at 7C20h.
4. Set AESST (ECON1<11>) to initiate the encryption.
5. Copy the plaintext message to TEXTB at 7C30h.
6. Wait for the hardware to clear AESST.
7. Read the ciphertext message from XOROUT at 7C40h.
8. Repeat steps 4 through 7 for subsequent blocks. The encryption output from the previous block automatically becomes the IV for the following block.

To decrypt a block using OFB mode:

1. Load the encryption key as described in **Section 15.3.1 "Key Support"**. Note that this mode does not make use of a decryption key.
2. Set AESOP<1:0> to '00'.
3. Copy the Initialization Value (IV) to TEXTA at 7C20h.
4. Set AESST to initiate the encryption.
5. Copy the ciphertext message to TEXTB at 7C30h.
6. Wait for the hardware to clear AESST.
7. Read the plaintext message from XOROUT at 7C40h.
8. Repeat steps 4 through 7 for subsequent blocks.

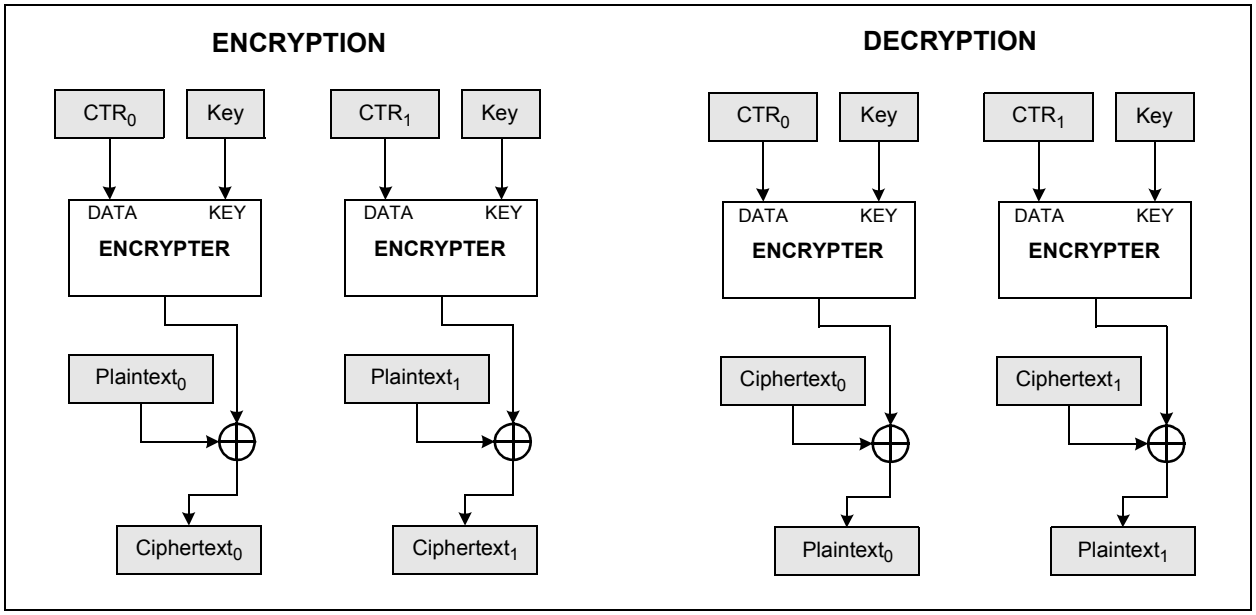
The context for OFB mode consists of both the AES encryption key and the encryption output from the most recent block. Save the encryption block output from TEXTA at 7C20h to be used as the IV when resuming the operation for additional blocks.

15.3.3.5 Counter Mode (CTR)

Counter mode is not directly supported by hardware, but can be implemented with software assistance. In CTR mode, a counter is used as the input to the encryption block. The encrypted output is then XORed with the plaintext to yield the ciphertext, or vice versa. The counter does not necessarily need to be a true counter; any practically non-repeating function will suffice. When using CTR mode, the application must load the counter value before each block.

Since each block depends on the counter value, identical subsequent plaintext blocks will yield different ciphertext blocks. Whether or not these blocks are independent will depend on the selected counter function. Figure 15-5 shows the use of CTR mode for encryption and decryption.

FIGURE 15-5: CTR ENCRYPTION AND DECRYPTION



To encrypt a block using CTR mode:

1. Load the encryption key as described in **Section 15.3.1 "Key Support"**.
2. Set AESOP<1:0> (ECON1<10:9>) to '00'.
3. Copy the counter value to TEXTA at 7C20h.
4. Set AESST (ECON1<11>) to initiate the encryption.
5. Copy the plaintext message to TEXTB at 7C30h.
6. Wait for the hardware to clear AESST.
7. Read the ciphertext message from XOROUT at 7C40h.
8. Repeat steps 3 through 7 for subsequent blocks.

To decrypt a block using CTR mode:

1. Load the encryption key as described in **Section 15.3.1 "Key Support"**. Note that this mode does not make use of a decryption key.
2. Set AESOP<1:0> to '00'.
3. Copy the counter value to TEXTA at 7C20h.
4. Set AESST to initiate the encryption.
5. Copy the ciphertext message to TEXTB at 7C30h.
6. Wait for the hardware to clear AESST.
7. Read the plaintext message from XOROUT at 7C40h.
8. Repeat steps 3 through 7 for subsequent blocks.

The context for CTR mode consists of the AES encryption key and the counter value. It is up to the application to determine what needs to be saved for the counter value context.

NOTES:

16.0 POWER-SAVING FEATURES

Due to the high bandwidth and long cable length requirements, Ethernet applications can utilize a significant amount of power. ENC424J600/624J600 devices include power-down and PHY power management features to assist low-power applications. While features cannot completely mitigate power requirements, they can help reduce power consumption when the Ethernet interface is not needed.

16.1 General Power-Down

The ENCX24J600 may be placed in Power-Down mode through the command interface. In this mode, the device will no longer be able to transmit or receive any packets or perform DMA operations. However, most registers, and all buffer memories, retain their states and remain accessible by the host controller. The clock driver also remains operational, leaving the CLKOUT function unaffected. However, the MAC/MII and PHY registers all become inaccessible, and the PHY registers lose their current states.

To power-down the Ethernet interface:

1. Turn off the Modular Exponentiation and AES engines by clearing CRYPTEN (EIR<15>).
2. Turn off packet reception by clearing RXEN (ECON1<0>).
3. Wait for any in-progress receptions to complete by polling RXBUSY (ESTAT<13>) until it is clear.
4. Wait for any current transmission operation to complete by verifying that TXRTS (ECON1<1>) is clear.
5. Power-down the PHY by setting the PSLEEP bit (PHCON1<11>).
6. Power-down the Ethernet interface by clearing ETHEN and STRCH (ECON2<15,14>). Disabling the LED stretching behavior is necessary to ensure no LEDs get trapped in a perpetually illuminated state in the event they are being stretched on when ETHEN is cleared.

To resume normal operation, the PHY registers need to be reconfigured after wake-up. The typical restart sequence is:

1. Wake-up the Ethernet interface by setting ETHEN and STRCH (ECON2<15,14>).
2. Wake-up the PHY by clearing PSLEEP (PHCON1<11>). Care should be taken to modify only the PSLEEP bit.
3. Restore receive capabilities by setting RXEN (ECON1<0>).

After leaving Sleep mode, there will be a delay of several hundred milliseconds before a new link is established. If the host controller attempts to transmit any Ethernet packets before the link is established, the PHY will suppress the transmission onto the wire to avoid interfering with auto-negotiation or violating IEEE 802.3 standards. The link status can be monitored through the Link Change Interrupt Flag, LINKIF (EIR<11>), and PHYLNK status bit (ESTAT<8>).

16.2 Energy Detect Power-Down

ENC424J600/624J600 devices also support an Energy Detect Power-Down mode. In this mode, the PHY remains powered down until a signal is detected on the Ethernet interface. While no packets can be sent or received, the internal PHY configuration is maintained. This is useful for applications in which the Ethernet cable may not always be connected, but need to automatically activate when a network cable is attached by the user and a link partner is detected.

When a signal is detected on the Ethernet medium, the EDSTAT flag (PHCON2<1>) is set.

To enable Energy Detect Power-Down mode, set the EDPWRDN bit (PHCON2<13>). The PHY automatically powers up and down based on the value of EDSTAT. When in Energy Detect Power-Down, the host microcontroller should monitor the Ethernet link status via the LINKIF interrupt flag and PHYLNK status bit. When linked, it should set ETHEN and STRCH (ECON2<15,14>) and begin using the network interface as normal. When unlinked, it should clear ETHEN and STRCH to save power. To resume normal operation, clear EDPWRDN. While the PHY is in Energy Detect Power-Down mode, the transmit logic will indefinitely hold off transmissions when unlinked. Therefore, if the application attempts to transmit a packet by setting TXRTS (ECON1<1>), this bit may not clear itself or cause a transmit interrupt to occur until the user plugs the device into another link partner.

A device in Energy Detect Power-Down mode does not transmit link pulses, but passively listens for the remote link partner to transmit a signal in order to wake the device. If the remote device is also in a similar Passive Listening mode, neither device will wake-up. This should not cause problems for normal Ethernet equipment, such as switches and routers, but may raise concerns if two embedded devices using this feature will be connected directly.

Note:	Through compatibility testing, it has been observed that some 3rd party Ethernet products do not transmit link pulses that are compliant with the IEEE 802.3 standard timing requirements. For such devices, the ENCX24J600 PHY energy detect feature may be unable to wake-up. Before enabling Energy Detect Power-Down, application designers should weigh the compatibility risks of using this feature, and where possible, implement a means of disabling it by the end product user.

16.3 External Power-Down

For applications that are extremely power-sensitive and have no need for memory or register retention, it may be useful to control the device using an external power-down circuit. This allows the host controller to completely remove power from the device. External power-down circuitry can be designed with either a MOSFET on the power supply pins, or by using a regulator with output enable capabilities.

Keep in mind that an externally controlled power-down will require the ENCX24J600 to be completely re-initialized, as described in **Section 8.0 “Initialization”**.

REGISTER 16-1: PHCON2: PHY CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
r	r	EDPWRDN	r	EDTHRES	r	r	r
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
r	r	r	r	r	FRCLNK	EDSTAT	r
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Reserved:** Write as '0', ignore on read

bit 13 **EDPWRDN:** Energy Detect Power-Down Enable bit
- 1 = Energy detect power-down enabled. PHY automatically powers up and down based on the state of EDSTAT.
- 0 = Energy detect power-down disabled. Use this setting for maximal compatibility.

bit 12 **Reserved:** Write as '0', ignore on read

bit 11 **EDTHRES:** Energy Detect Threshold Control bit
- 1 = Less energy is required to wake the PHY from energy detect power-down
- 0 = Normal energy detect threshold

bit 10-3 **Reserved:** Write as '0', ignore on read

bit 2 **FRCLNK:** Force Link Control bit
- 1 = Force immediate link up, even when no link partner is present (100 Mbps operation only)(1)
- 0 = Normal operation

bit 1 **EDSTAT:** Energy Detect Status bit
- 1 = Energy detect circuit has detected energy on the TPIN+/- pins within the last 256 ms
- 0 = No energy has been detected on the TPIN+/- pins within the last 256 ms

bit 0 **Reserved:** Write as '0', ignore on read

Note 1: Intended for testing purposes only. Do not use in 10 Mbps operation.

NOTES:

17.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

Storage temperature ..... -65°C to +150°C
Ambient temperature under bias ..... -40°C to +85°C (Industrial)
Voltage on V_{DD}, V_{DDOSC}, V_{DDPLL}, V_{DDRX} and V_{DDTX}, with respect to V_{SS} ..... -0.3V to 4.0V
Voltage on any digital pin, with respect to V_{SS} ..... -0.3V to 6.0V
Voltage on OSC1 and RBIAS analog pins, with respect to V_{SS} ..... -0.3V to V_{DD} + 0.3V
Voltage on TPIN+/- and TPOUT+/-, with respect to V_{SS} ..... -0.3V to 5.0V
Voltage on V_{CAP}, with respect to all V_{SS} pins **(Note 1)** ..... -0.3V to 2.0V
ESD protection on all pins ..... 2 kV
Current sourced or sunk by any digital output pin ..... 25 mA
Current out of all V_{SS} pins ..... 420 mA
Current into all V_{DD} pins ..... 300 mA

Note 1: V_{CAP} is not designed to supply an external load. No external voltage should be applied to this pin.

† **Notice:** Stresses above those listed under “Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

17.1 DC Characteristics: ENC424J600/624J600

TABLE 17-1: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min	Typ	Max	Unit
ENC424J600/624J600:					
Operating Junction Temperature Range	T_J	-40	—	+125	°C
Operating Ambient Temperature Range	T_A	-40	—	+85	°C
Power Dissipation: Internal Chip Power Dissipation: $P_{INT} = V_{DD} \times (I_{DD} - \Sigma I_{OH})$ I/O Pin Power Dissipation: $P_{I/O} = \Sigma (\{V_{DD} - V_{OH}\} \times I_{OH}) + \Sigma (V_{OL} \times I_{OL}) + ((V_{TPOUT+} + V_{TPOUT-})/2 \times I_{TXCT})$	P_D	$P_{INT} + P_{I/O}$			W
Maximum Allowed Power Dissipation	P_{DMAX}	$(T_J - T_A)/\theta_{JA}$			W

TABLE 17-2: THERMAL PACKAGING CHARACTERISTICS

Characteristic	Symbol	Typ	Max	Unit	Notes
Package Thermal Resistance, 44-Pin QFN (8x8x1 mm)	θ_{JA}	28	—	°C/W	**(Note 1)**
Package Thermal Resistance, 44-Pin TQFP (10x10x1 mm)	θ_{JA}	49.8	—	°C/W	**(Note 1)**
Package Thermal Resistance, 64-Pin TQFP (10x10x1 mm)	θ_{JA}	47	—	°C/W	**(Note 1)**

Note 1: Junction to ambient thermal resistance, Theta-$_{JA}$ (θ_{JA}) numbers are achieved by package simulations.

TABLE 17-3: DC CHARACTERISTICS: ENC424J600/624J600 (INDUSTRIAL)

DC CHARACTERISTICS			Standard Operating Conditions: -40°C ≤ TA ≤ 85°C, 3.0V ≤ VDD ≤ 3.6V (Industrial)				
Param. No.	**Sym**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
D001	VDD	**Supply Voltage**	3.0	3.3	3.6	V	
D002	VPOR	**VDD Start Voltage** to Ensure Internal Power-on Reset Signal	1.75	—	1.95	V	
D003	SVDD	**VDD Rise Rate** to Ensure Internal Power-on Reset Signal	0.05	—	—	V/ms	
	VIH	**Input High Voltage**				V	
D004		Digital Input Pins	0.6 VDD	—	5.5	V	
D005		OSC1 Pin	0.7 VDD	—	VDD	V	
	VIL	**Input Low Voltage**					
D006		Digital Input Pins	VSS	—	0.2 VDD	V	
D007		OSC1 Pin	VSS	—	0.2 VDD	V	
	VOH	**Output High Voltage**					
		All Digital Output Pins	2.4	—	VDD	V	IOH = -8mA
	VOL	**Output Low Voltage**					
		All Digital Output Pins	VSS	—	0.4	V	IOL = 8mA
	IWPU	**Weak Pull-up Current**	-150	-260	-400	µA	VDD = 3.3V, VPIN = VSS
	IWPD	**Weak Pull-Down Current**	28	56	112	µA	VDD = 3.3V, VPIN = VDD
	IIL	**Input Leakage Current**					
		Digital Input Pins	—	—	±1	µA	VSS ≤ VPIN ≤ VDD
		OSC1 Pin	—	—	±150	µA	VSS ≤ VPIN ≤ VDD
	IDD	**Supply Current**					
		Not Linked	—	74	—	mA	**(Note 1)**
		100Base-TX Linked, Idle	—	96	—	mA	**(Note 1)**
		100Base-TX Linked, Maximum TX Utilization	—	96	117	mA	**(Note 1)**
		10Base-T Linked, Idle	—	82	—	mA	**(Note 1)**
		10Base-T Linked, Maximum TX Utilization	—	82	103	mA	**(Note 1)**
	ΔICT	**Cryptographic Module Current**	—	40	—	mA	EIR<15> = 1
	ITXCT	**TX Transformer Center Tap Current**					
		Not Linked	—	1	—	mA	
		100Base-TX Linked	—	30	—	mA	
		10Base-T Linked	—	80	—	mA	

Note 1: Excludes TX transformer center tap and LEDA/LEDB currents; cryptographic engine module disabled (EIR<15> = 0).

2: Cryptographic engine module disabled (EIR<15> = 0), auto-negotiation disabled (PHCON1<12> = 0) and Ethernet disabled (ECON2<15> = 0).

3: Measured across 100Ω termination on cable side of transformer.

TABLE 17-3: DC CHARACTERISTICS: ENC424J600/624J600 (INDUSTRIAL) (CONTINUED)

DC CHARACTERISTICS			Standard Operating Conditions: -40°C ≤ TA ≤ 85°C, 3.0V ≤ VDD ≤ 3.6V (Industrial)				
Param. No.	**Sym**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
	IPD	**Power-Down Current(2)**					
		Energy Detect Power-Down	—	29.2	34	mA	PHCON2<13> = 1
		Sleep	—	23.8	28	mA	PHCON1<11> = 1
	VTPOUT	**Peak Differential Output Voltage**				V	
		100Base-TX	0.95	1.00	1.05	V	**(Note 3)**
		10Base-T	2.2	2.5	2.8	V	**(Note 3)**
	VSQ	**10Base-T RX Differential Squelch Threshold**	160	300	585	mV	

Note 1: Excludes TX transformer center tap and LEDA/LEDB currents; cryptographic engine module disabled (EIR<15> = 0).

2: Cryptographic engine module disabled (EIR<15> = 0), auto-negotiation disabled (PHCON1<12> = 0) and Ethernet disabled (ECON2<15> = 0).

3: Measured across 100Ω termination on cable side of transformer.

TABLE 17-4: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

Operating Conditions: -40°C < TA < +85°C (unless otherwise stated)							
Param No.	**Symbol**	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
	VRGOUT	Regulator Output Voltage	1.62	1.80	1.98	V	
	CEFC	External Filter Capacitor Value	4.7	10	—	μF	Capacitor must have low series resistance (< 3Ω)

TABLE 17-5: REQUIREMENTS FOR EXTERNAL MAGNETICS

Parameter	Min	Norm	Max	Units	Conditions
RX Transformer Turns Ratio	—	1:1	—	—	
TX Transformer Turns Ratio	—	1:1	—	—	Transformer Center Tap = 3.3V
Insertion Loss	—	—	1.1	dB	
Primary Inductance	350	—	—	μH	8 mA bias
Transformer Isolation	—	1.5	—	kV	
Differential to Common-Mode Rejection	40	—	—	dB	0.1 to 10 MHz
Return Loss	-16	—	—	dB	

17.2 AC Characteristics: ENC424J600/624J600 (Industrial)

AC CHARACTERISTICS	Standard Operating Conditions -40°C ≤ TA ≤ +85°C, 3.00V ≤ VDD ≤ 3.60V (Industrial)

TABLE 17-6: OSCILLATOR TIMING CHARACTERISTICS

Param. No.	Sym	Characteristic	Min	Max	Units	Conditions
	FOSC	Clock In Frequency	25	25	MHz	
	TOSC	Clock In Period	40	40	ns	
	TDUTY	Duty Cycle (external clock input)	40	60	%	
	Δf	Clock Frequency Error	—	±50	ppm	

TABLE 17-7: CLKOUT PIN TIMING SPECIFICATIONS

Param. No.	Sym	Characteristic	Min	Typ	Max	Units	Conditions
	FCLKOUT	CLKOUT Frequency	DC	—	33.3	MHz	
	TDUTY	CLKOUT Duty Cycle	40	50	60	%	All prescaler settings except divide by 12.5
			37.5	47.5	57.5	%	Divide by 12.5 prescaler

TABLE 17-8: CLKOUT PIN AC CHARACTERISTICS

Param. No.	Sym	Characteristic	Min	Max	Units	Conditions
	t_{r}CLKOUT	CLKOUT Pin Rise Time	—	3	ns	**(Note 1)**
	t_{f}CLKOUT	CLKOUT Pin Fall Time	—	4	ns	**(Note 1)**
	ΔCLKOUT	CLKOUT Stability (jitter)	—	±0.25	%	

Note 1: Measured from 0.1 VDD to 0.9 VDD with a load of 10 pF.

FIGURE 17-1: SPI INPUT TIMING

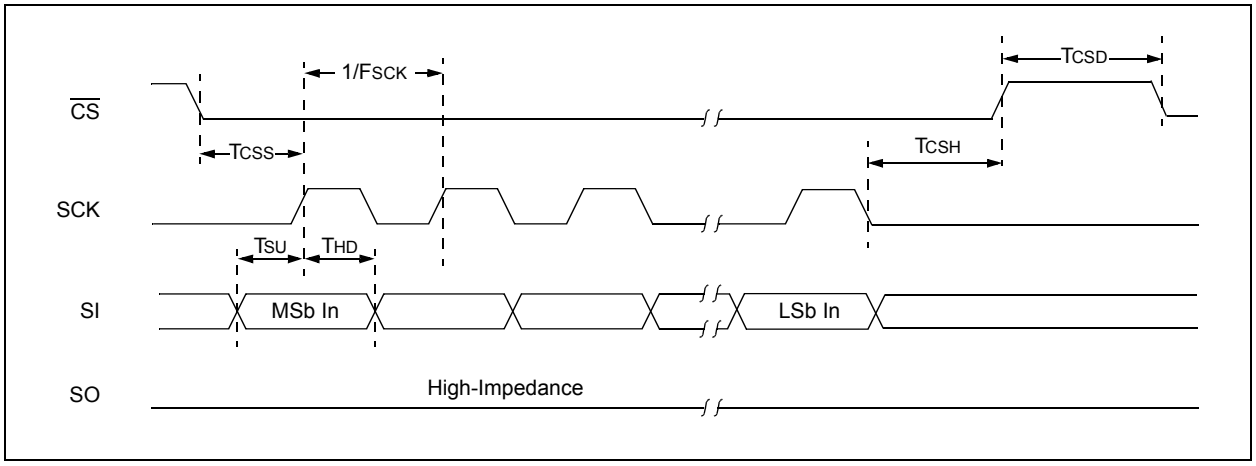


FIGURE 17-2: SPI OUTPUT TIMING

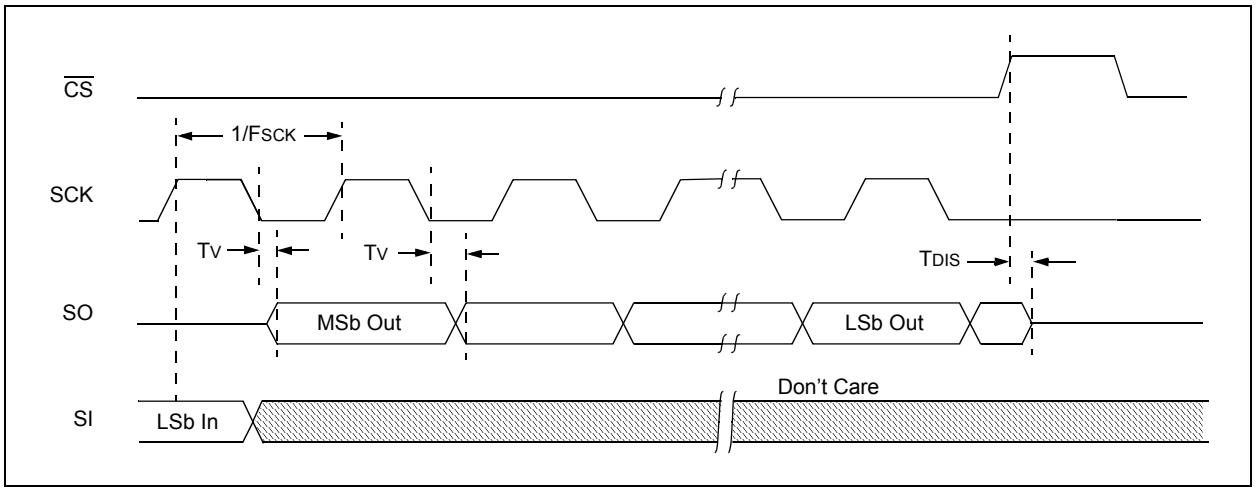


TABLE 17-9: SPI INTERFACE AC CHARACTERISTICS

Sym	Characteristic	Min	Typ	Max	Units	Conditions
FSCK	SPI Clock Frequency	DC	—	14	MHz	
TDUTY	SCK Duty Cycle	45	—	55	%	
TCSS	$\overline{CS}$ Setup Time	50	—	—	ns	
TCSH	$\overline{CS}$ Hold Time	50	—	—	ns	
TCSD	$\overline{CS}$ Disable Time	20	—	—	ns	
TSU	Data Setup Time	10	—	—	ns	
THD	Data Hold Time	10	—	—	ns	
TV	Output Valid from Clock Low	—	—	10	ns	Load on SO pin = 30 pF
TDIS	Output Disable Time	—	—	10	ns	Load on SO pin = 30 pF

TABLE 17-10: PSP INTERFACE TIMING SPECIFICATIONS

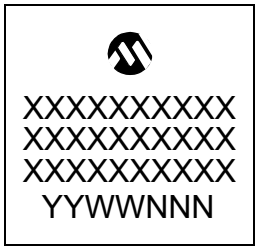
Sym	Characteristic	Min	Typ	Max	Units	Comments
TPSP1	CS, Address, R/$\overline{W}$ Setup Time	1	—	—	ns	
TPSP2	RD, EN, BxSEL to Data Valid	—	—	75	ns	
TPSP3	Data Output Hold Time	0	—	3	ns	
TPSP4	RD, EN, BxSEL Deassertion Time	4.5	—	—	ns	
TPSP5	CS, R/$\overline{W}$ Setup Time	3.5	—	—	ns	
TPSP6	Address Setup Time	3.5	—	—	ns	
TPSP7	Data Setup Time	3.5	—	—	ns	
TPSP8	WR, WRL, WRH, EN, BxSEL Assertion Time	6.5	—	—	ns	
TPSP9	Address Hold Time	1	—	—	ns	
TPSP10	Data Input Hold Time	1	—	—	ns	
TPSP11	WR, WRL, WRH, EN, BxSEL Deassertion Time	4.5	—	—	ns	SFR access
		40	—	—	ns	SRAM access
TPSP12	CS, Address Setup Time	6.5	—	—	ns	
TPSP13	AL Assertion Time	6.5	—	—	ns	
TPSP14	Address Hold Time	1	—	—	ns	
TPSP15	AL Deassertion Time	4	—	—	ns	

NOTES:

18.0 PACKAGING INFORMATION

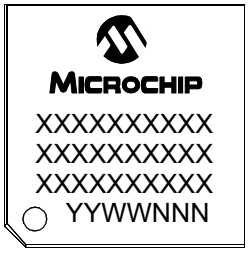
18.1 Package Marking Information

44-Lead QFN

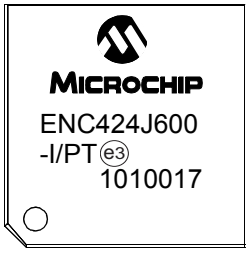


Example

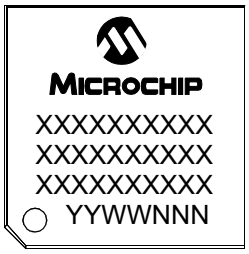
44-Lead TQFP



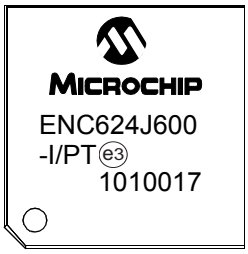
Example



64-Lead TQFP (10x10x1mm)



Example



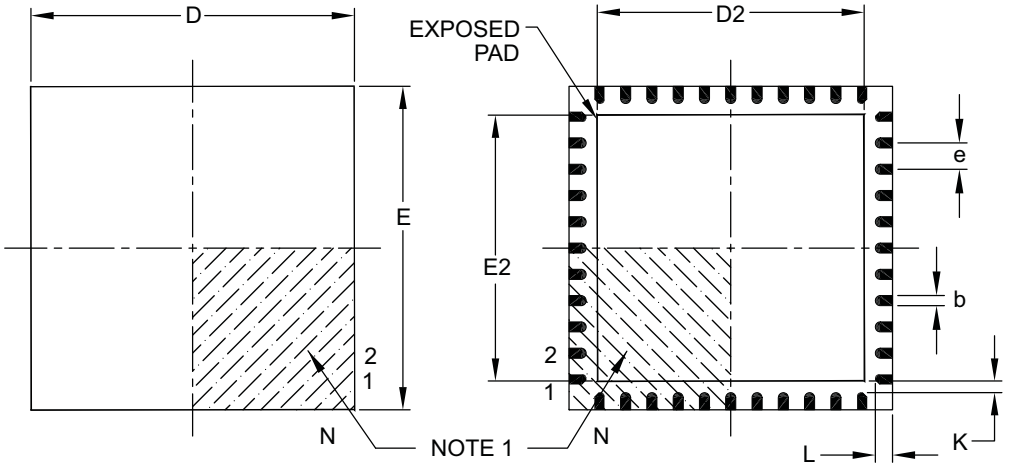
Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
		Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.	

18.2 Package Details

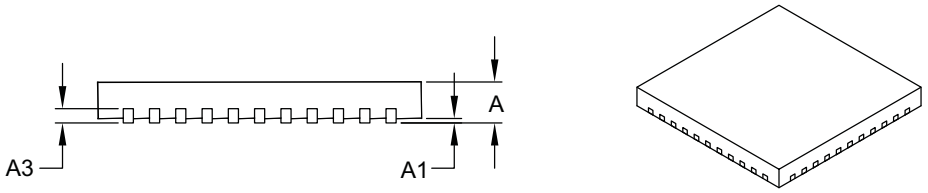
The following sections give the technical details of the packages.

44-Lead Plastic Quad Flat, No Lead Package (ML) – 8x8 mm Body [QFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW **BOTTOM VIEW**



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		44	
Pitch	e		0.65 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.20 REF	
Overall Width	E		8.00 BSC	
Exposed Pad Width	E2	6.30	6.45	6.80
Overall Length	D		8.00 BSC	
Exposed Pad Length	D2	6.30	6.45	6.80
Contact Width	b	0.25	0.30	0.38
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	–	–

Notes:

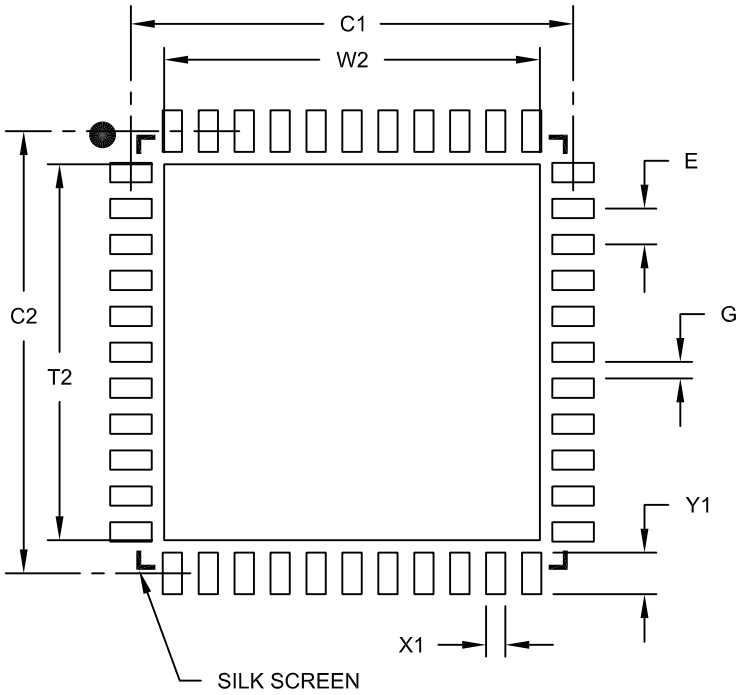
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-103B

44-Lead Plastic Quad Flat, No Lead Package (ML) – 8x8 mm Body [QFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			6.80
Optional Center Pad Length	T2			6.80
Contact Pad Spacing	C1		8.00	
Contact Pad Spacing	C2		8.00	
Contact Pad Width (X44)	X1			0.35
Contact Pad Length (X44)	Y1			0.80
Distance Between Pads	G	0.25		

Notes:

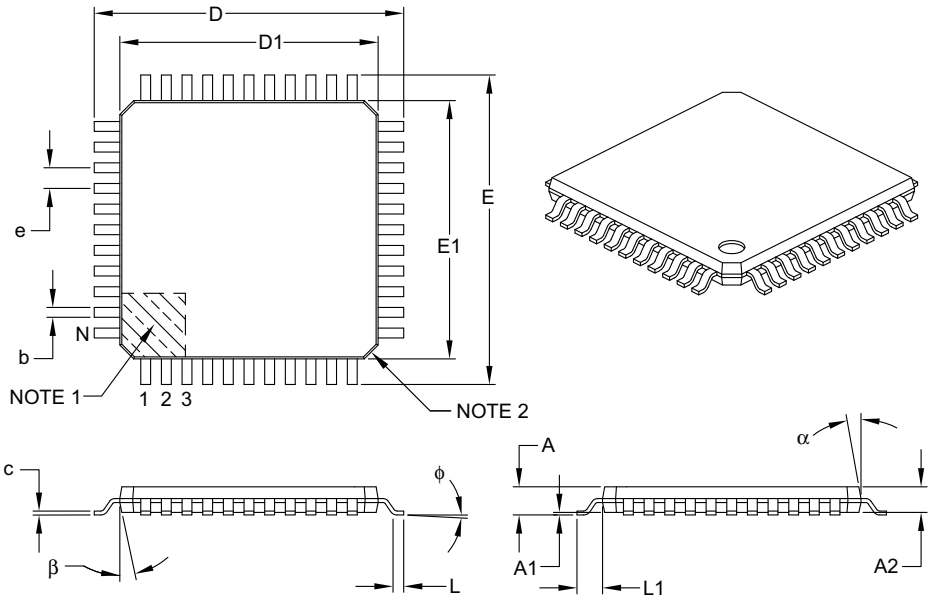
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2103A

44-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads	N		44	
Lead Pitch	e		0.80 BSC	
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	–	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Foot Angle	ϕ	0°	3.5°	7°
Overall Width	E		12.00 BSC	
Overall Length	D		12.00 BSC	
Molded Package Width	E1		10.00 BSC	
Molded Package Length	D1		10.00 BSC	
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.30	0.37	0.45
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

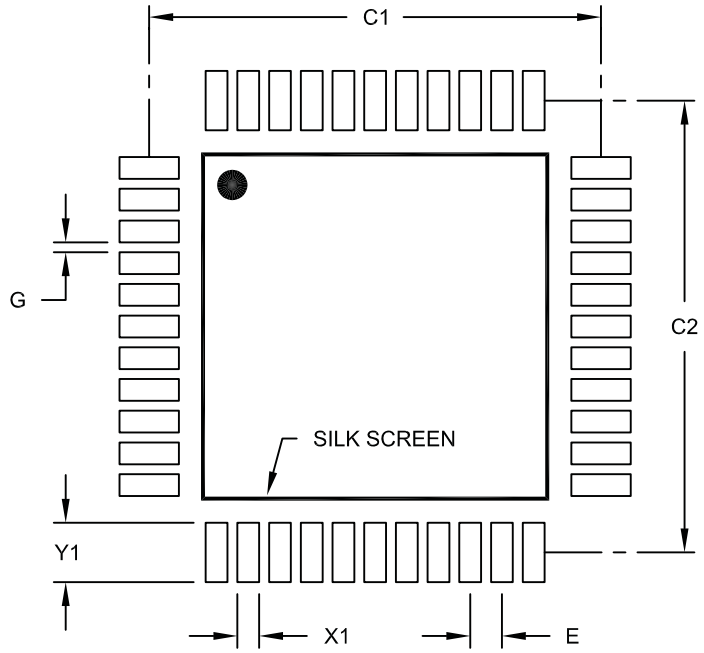
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-076B

44-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.80 BSC	
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X44)	X1			0.55
Contact Pad Length (X44)	Y1			1.50
Distance Between Pads	G	0.25		

Notes:

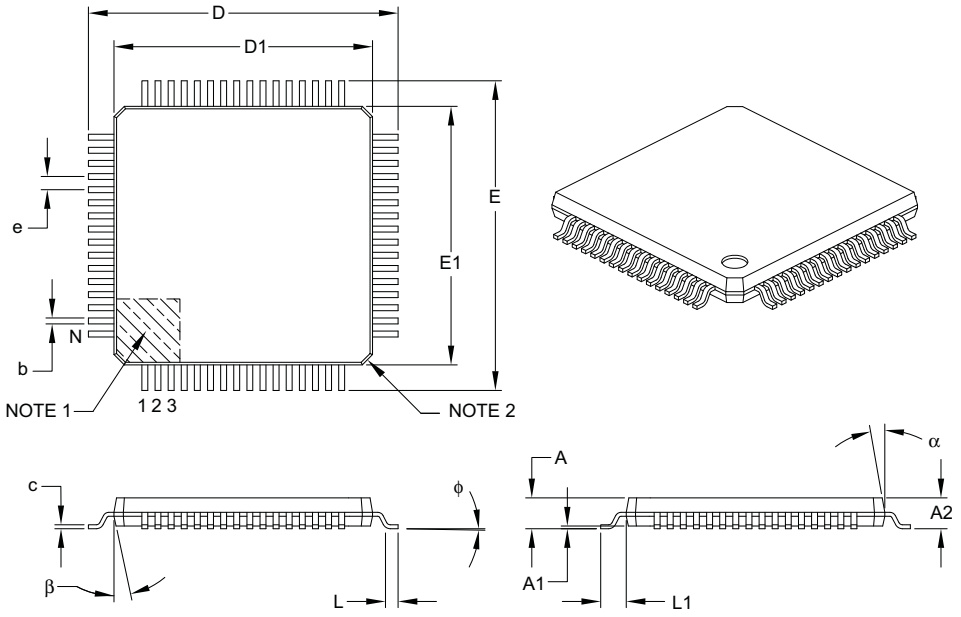
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2076A

64-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads	N		64	
Lead Pitch	e		0.50 BSC	
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	–	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Foot Angle	ϕ	0°	3.5°	7°
Overall Width	E		12.00 BSC	
Overall Length	D		12.00 BSC	
Molded Package Width	E1		10.00 BSC	
Molded Package Length	D1		10.00 BSC	
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

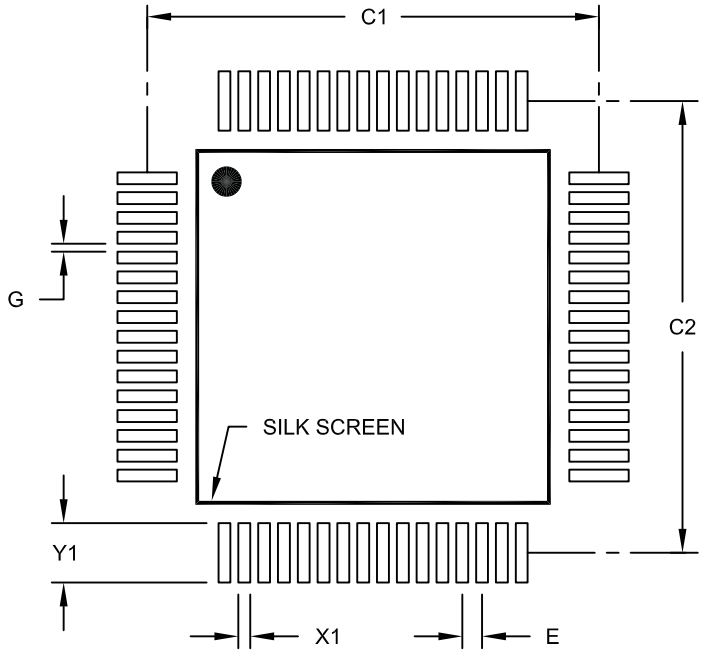
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-085B

64-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X64)	X1			0.30
Contact Pad Length (X64)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2085A

NOTES:

APPENDIX A: REVISION HISTORY

Revision A (March 2009)

Original data sheet for ENC424J600/624J600 devices.

Revision B (July 2009)

Removed preliminary from the data sheet. **Section 1.0 "Device Overview"** and **Section 7.0 "Reset"** had minor edits.

Revision C (January 2010)

Section 5.3.3 "MODE 3" and **Section 5.3.4 "MODE 4"** had minor edits.

NOTES:

INDEX

S

T

W

NOTES:

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com, click on Customer Change Notification and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://support.microchip.com

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: ENC424J600/624J600 Literature Number: DS39935C

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	X	/XX	XXX
Device	Temperature Range	Package	Pattern

Device	ENC424J600, ENC624J600, ENC424J600T(1), ENC624J600T(1); VDD range 3.0V to 3.6V
Temperature Range	I = -40°C to +85°C (Industrial)
Package	ML = QFN (Quad Flat No Lead) PT = TQFP (Thin Quad Flatpack)
Pattern	Three-Digit Code or Special Requirements (blank otherwise) ES = Engineering Sample

Examples:

a) ENC424J600-I/ML = Industrial temp., QFN package.

b) ENC424J600-I/PT = Industrial temp., 44 leads TQFP package.

c) ENC624J600T-I/PT = Industrial temp., 64 leads TQFP package, tape and reel.

Note 1: T = in tape and reel.

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-6578-300
Fax: 886-3-6578-370

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

01/05/10

LifeElectronics
Живое партнерство

ООО «ЛайфЭлектроникс» **“LifeElectronics” LLC**

ИНН 7805602321 КПП 780501001 Р/С 40702810122510004610 ФАКБ "АБСОЛЮТ БАНК" (ЗАО) в г.Санкт-Петербурге К/С 30101810900000000703 БИК 044030703

Компания «Life Electronics» занимается поставками электронных компонентов импортного и отечественного производства от производителей и со складов крупных дистрибьюторов Европы, Америки и Азии.

С конца 2013 года компания активно расширяет линейку поставок компонентов по направлению коаксиальный кабель, кварцевые генераторы и конденсаторы (керамические, пленочные, электролитические), за счёт заключения дистрибьюторских договоров

Мы предлагаем:

- *Конкурентоспособные цены и скидки постоянным клиентам.*
- *Специальные условия для постоянных клиентов.*
- *Подбор аналогов.*
- *Поставку компонентов в любых объемах, удовлетворяющих вашим потребностям.*

- *Приемлемые сроки поставки, возможна ускоренная поставка.*
- *Доставку товара в любую точку России и стран СНГ.*
- *Комплексную поставку.*
- *Работу по проектам и поставку образцов.*
- *Формирование склада под заказчика.*

- *Сертификаты соответствия на поставляемую продукцию (по желанию клиента).*
- *Тестирование поставляемой продукции.*
- *Поставку компонентов, требующих военную и космическую приемку.*
- *Входной контроль качества.*
- *Наличие сертификата ISO.*

В составе нашей компании организован Конструкторский отдел, призванный помогать разработчикам, и инженерам.

Конструкторский отдел помогает осуществить:

- *Регистрацию проекта у производителя компонентов.*
- *Техническую поддержку проекта.*
- *Защиту от снятия компонента с производства.*
- *Оценку стоимости проекта по компонентам.*
- *Изготовление тестовой платы монтаж и пусконаладочные работы.*

Тел: +7 (812) 336 43 04 (многоканальный)
Email: org@lifeelectronics.ru

www.lifeelectronics.ru