Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	EBI/EMI, Ethernet, I²C, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	70
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.375V ~ 3.465V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12ne64cpve

A.18 External Bus Timing 541

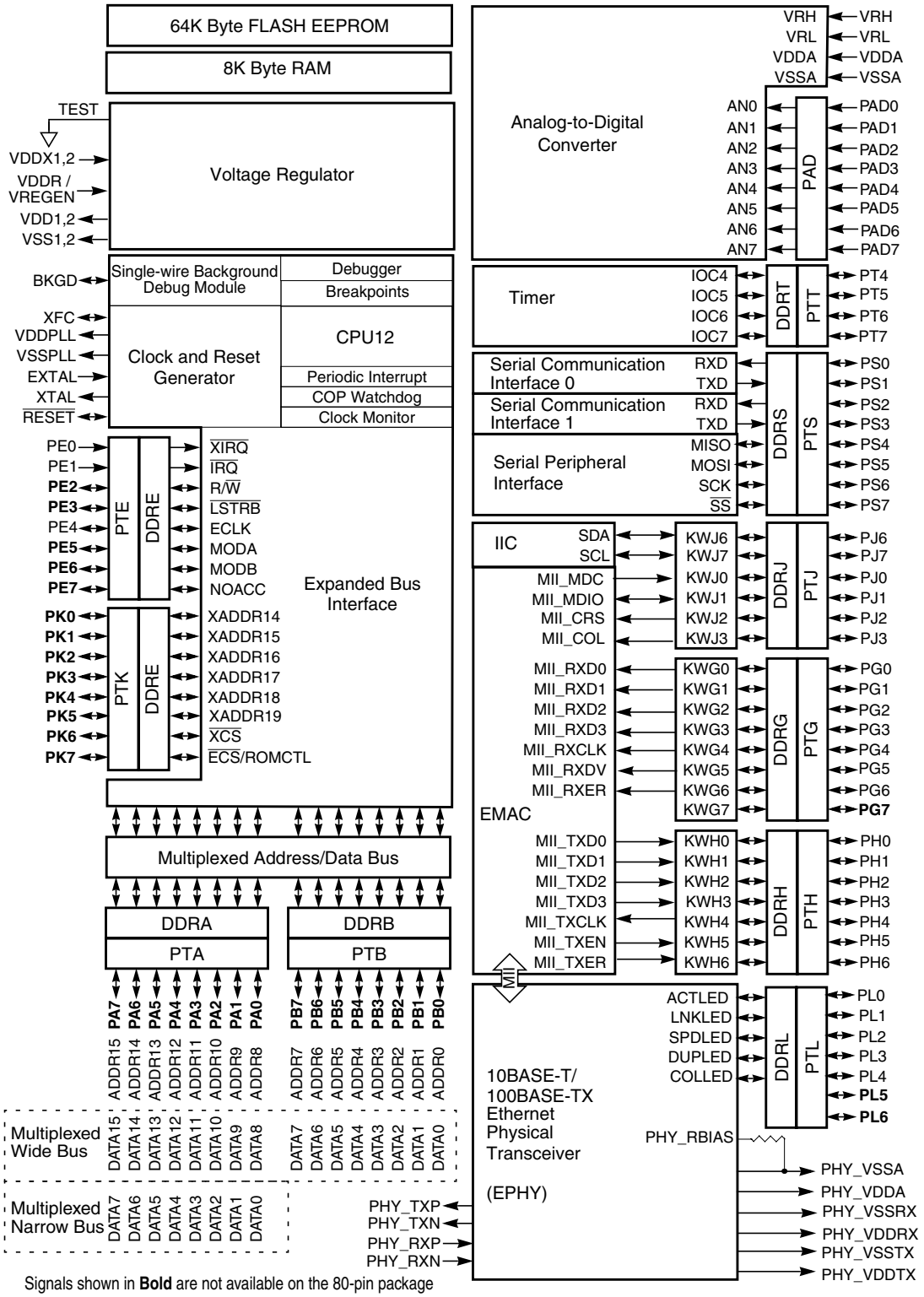
Appendix B
Schematic and PCB Layout Design Recommendations

B.1 Introduction 543
B.1.1 Schematic Designing with the MC9S12NE64 and Adding an Ethernet Interface 543
B.1.2 Power Supply Notes 545
B.1.3 Clocking Notes 545
B.1.4 EPHY Notes 545
B.1.5 EPHY LED Indicator Notes 545
B.2 PCB Design Recommendation 546
B.2.1 General PCB Design Recommendations 546
B.2.2 Ethernet PCB Design Recommendations 546

Appendix C
Package Information

C.1 112-Pin LQFP Package 549
C.2 80-Pin TQFP-EP Package 550

1.1.3 Block Diagram



Signals shown in **Bold** are not available on the 80-pin package

Figure 1-1. MC9S12NE64 Block Diagram

This figure shows a suggested map, which is not the map out of reset. After reset the map is:

$0000 – $03FF: register space
$0000 – $1FFF: 7K RAM (1K RAM hidden behind register space)

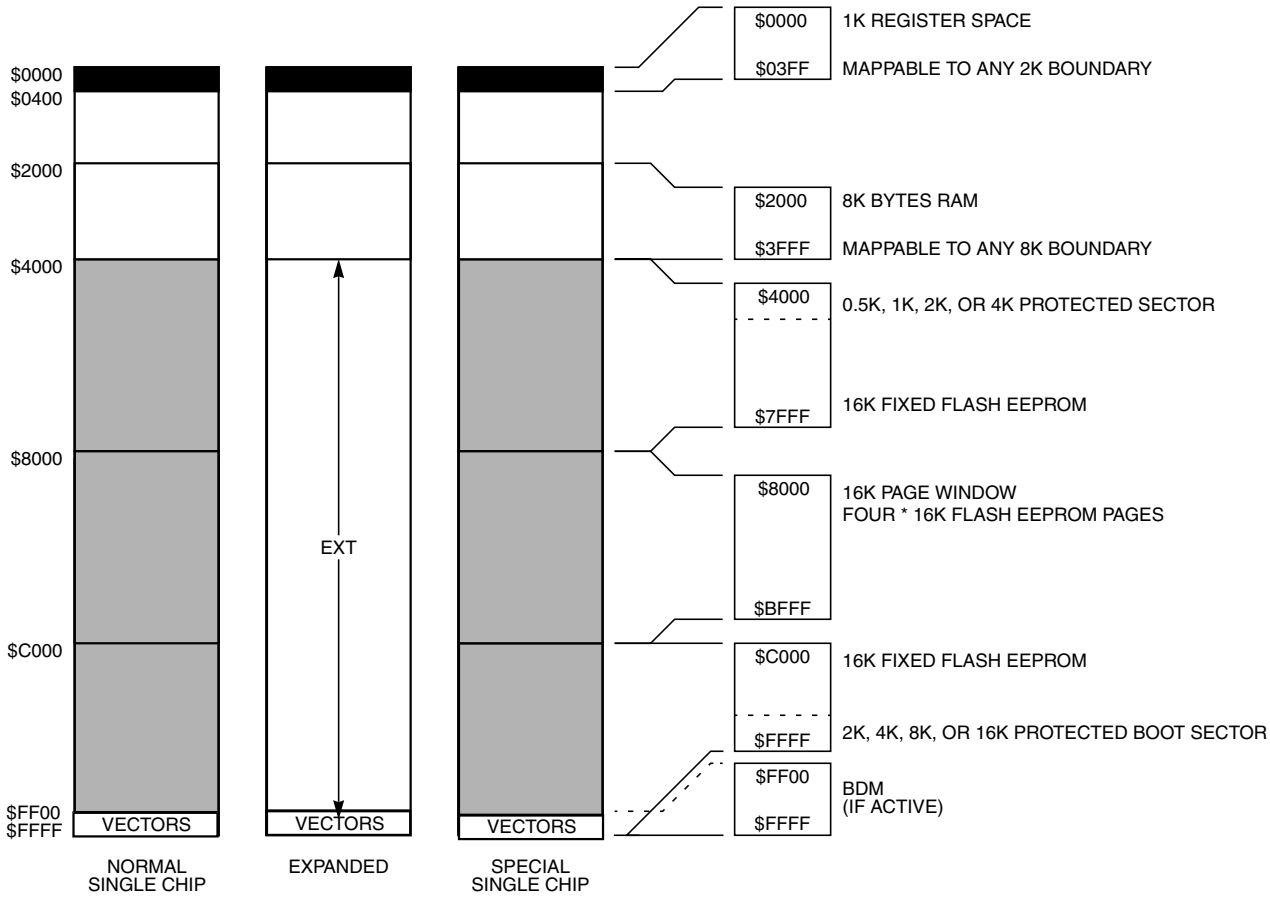


Figure 1-2. MC9S12NE64 User Configurable Memory Map

1.1.5 Detailed Register Map

The following tables show the register maps of the MC9S12NE64. For detailed information about register functions, please see the appropriate block description chapter.

$0140 - $016F Ethernet Media Access Controller (EMAC)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$0140	NETCTL	Read:	EMACE	0	0	ESWAI	EXTPHY	MLB	FDX	0
		Write:								
$0141	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								
$0142	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								
$0143	RXCTS	Read:	RXACT	0	0	RFCE	0	PROM	CONMC	BCREJ
		Write:								
$0144	TXCTS	Read:	TXACT	0	CSLF	PTRC	SSB	0	0	0
		Write:							TCMD	
$0145	ETCTL	Read:	FPET	0	0	FEMW	FIPV6	FARP	FIPV4	FIEEE
		Write:								
$0146	ETYPE	Read:	ETYPE[15:8]							
		Write:								
$0147	ETYPE	Read:	ETYPE[7:0]							
		Write:								
$0148	PTIME	Read:	PTIME[15:8]							
		Write:								
$0149	PTIME	Read:	PTIME[7:0]							
		Write:								
$014A	IEVENT [15:8]	Read:	RFCIF	0	BREIF	RXEIF	RXAOIF	RXBOIF	RXACIF	RXBCIF
		Write:								
$014B	IEVENT [7:0]	Read:	MMCIF	0	LCIF	ECIF	0	0	TXCIF	0
		Write:								
$0141C	IMASK [15:8]	Read:	RFCIE	0	BREIE	RXEIE	RXAOIE	RXBOIE	RXACIE	RXBCIE
		Write:								
$014D	IMASK [7:0]	Read:	MMCIE	0	LCIE	ECIE	0	0	TXCIE	0
		Write:								
$014E	SWRST	Read:	0	0	0	0	0	0	0	0
		Write:	MACRST							
$014F	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								
$0150	MPADR	Read:	0	0	0	PADDR				
		Write:								
$0151	MRADR	Read:	0	0	0	RADDR				
		Write:								
$0152	MWDATA	Read:	WDATA[15:8]							
		Write:								
$0123	MWDATA	Read:	WDATA[7:0]							
		Write:								
$0154	MRDATA	Read:	RDATA[15:8]							
		Write:								
$0155	MRDATA	Read:	RDATA[7:0]							
		Write:								

1.7.5.3.2 Stop

During system low-power stop mode, the EMAC is immediately disabled. Any receive in progress is dropped and any PAUSE time-out is cleared. The user must not to enter low-power stop mode when TXACT or BUSY are set.

1.7.6 Ethernet Physical Transceiver (EPHY)

See the EPHY chapter for information about the Ethernet physical transceiver module. The EPHY also has MII register space which is not part of the MCU address space and not accessible via the IP bus. The MII registers can be accessed using the MDIO functions of the EMAC when the EMAC is configured for internal PHY operation. The MII pins of the EPHY are not externally accessible. All communication and management of the EPHY must be performed using the EMAC.

The organization unique identifier (OUI) for the MC9S12NE64 is 00-60-11 (hex).

1.7.6.1 Low-Power Operation

Special care must be taken when executing STOP and WAIT instructions while using the EPHY or undesired operation may result.

1.7.6.1.1 Wait

Transmit and receive operations are not possible in wait mode if the CWAI bit is set in the CLKSEL register because the clocks to the internal MII interface are stopped.

1.7.6.1.2 Stop

During system low-power stop mode, the EPHY is immediately reset and powered down. Upon exiting stop mode, the a start-up delay is required prior to initiating MDIO communications with the EPHY. See A.14, “EPHY Electrical Characteristics” for details. It is not possible to use an EPHY interrupt to wake the system from stop mode.

1.7.7 RAM 8K Block Description

This module supports single-cycle misaligned word accesses without wait states.

In addition to operating as the CPU storage, the 8K system RAM also functions as the Ethernet buffer while the EMAC module is enabled. While the EMAC is enabled, the Ethernet buffer will occupy 0.375K to 4.5K of RAM with physical addresses starting at $0000 and ending at $017F up to $11FF, depending on the setting of the BUFMAP bits in the EMAC Ethernet buffer configuration register (BUFCFG). The relative RAM address, which are controlled by settings in the internal RAM position register (INTRM), must be tracked in software.

The Ethernet buffer operation of the RAM is independent of the CPU and allows same cycle read/write access from the CPU and the EMAC. No hardware blocking mechanism is implemented to prevent the CPU from accessing the Ethernet RAM space, so care must be taken to ensure that the CPU does not corrupt the RAM Ethernet contents.

KEYACC bit is set, a write to a backdoor key address in the Flash memory triggers a comparison between the written data and the backdoor key data stored in the Flash memory. If all four words of data are written to the correct addresses in the correct order and the data matches the backdoor keys stored in the Flash memory, the MCU will be unsecured. The data must be written to the backdoor keys sequentially starting with 0xFF00–0xFF01 and ending with 0xFF06–0xFF07. 0x0000 and 0xFFFF are not permitted as backdoor keys. While the KEYACC bit is set, reads of the Flash memory will return invalid data.

The user code stored in the Flash memory must have a method of receiving the backdoor key from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN[1:0] bits are in the enabled state (see Section 2.3.2.2, “Flash Security Register (FSEC)”), the MCU can be unsecured by the backdoor access sequence described below:

1. Set the KEYACC bit in the Flash configuration register (FCNFG).
2. Write the correct four 16-bit words to Flash addresses 0xFF00–0xFF07 sequentially starting with 0xFF00.
3. Clear the KEYACC bit.
4. If all four 16-bit words match the backdoor keys stored in Flash addresses 0xFF00–0xFF07, the MCU is unsecured and the SEC[1:0] bits in the FSEC register are forced to the unsecure state of 1:0.

The backdoor key access sequence is monitored by an internal security state machine. An illegal operation during the backdoor key access sequence will cause the security state machine to lock, leaving the MCU in the secured state. A reset of the MCU will cause the security state machine to exit the lock state and allow a new backdoor key access sequence to be attempted. The following operations during the backdoor key access sequence will lock the security state machine:

1. If any of the four 16-bit words does not match the backdoor keys programmed in the Flash array.
2. If the four 16-bit words are written in the wrong sequence.
3. If more than four 16-bit words are written.
4. If any of the four 16-bit words written are 0x0000 or 0xFFFF.
5. If the KEYACC bit does not remain set while the four 16-bit words are written.
6. If any two of the four 16-bit words are written on successive MCU clock cycles.

After the backdoor keys have been correctly matched, the MCU will be unsecured. After the MCU is unsecured, the Flash security byte can be programmed to the unsecure state, if desired.

In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0xFF00–0xFF07 in the Flash configuration field.

The security as defined in the Flash security byte (0xFF0F) is not changed by using the backdoor key access sequence to unsecure. The backdoor keys stored in addresses 0xFF00–0xFF07 are unaffected by the backdoor key access sequence. After the next reset of the MCU, the security state of the Flash module is determined by the Flash security byte (0xFF0F). The backdoor key access sequence has no effect on the program and erase protections defined in the Flash protection register.

It is not possible to unsecure the MCU in special single-chip mode by using the backdoor key access sequence via the background debug mode (BDM).

Table 3-1. Pin Functions and Priorities (Sheet 3 of 4)

Port	Pin Name	Pin Function	Description	Pin Function after Reset
Port J	PJ[7]	IIC_SCL	Serial Clock Line bidirectional pin of IIC module	GPIO
		KWU/GPIO	Key board wake up Interrupt or General-purpose I/O	
	PJ[6]	IIC_SDA	Serial Data Line bidirectional pin of IIC module	
		KWU/GPIO	Key board wake up Interrupt or General-purpose I/O	
	PJ[3]	MII_COL	MII Collision	
		KWU/GPIO	Key board wake up Interrupt or General-purpose I/O	
	PJ[2]	MII_CRS	MII Carrier Sense	
		KWU/GPIO	Key board wake up Interrupt or General-purpose I/O	
	PJ[1]	MII_MDIO	MII Management Data Input/Output	
		KWU/GPIO	Key board wake up Interrupt or General-purpose I/O	
	PJ[0]	MII_MDC	MII Management Data Clock	
		KWU/GPIO	Key board wake up Interrupt or General-purpose I/O	
Port L	PL[6]	GPIO	General-purpose I/O	GPIO
	PL[5]	GPIO	General-purpose I/O	
	PL[4]	COLLED	EPHY Collision LED	
		GPIO	General-purpose I/O	
	PL[3]	DUPLED	EPHY Duplex LED	
		GPIO	General-purpose I/O	
	PL[2]	SPDLED	EPHY Speed LED	
		GPIO	General-purpose I/O	
	PL[1]	LNKLED	EPHY Link LED	
		GPIO	General-purpose I/O	
	PL[0]	ACTLED	EPHY Active LED	
		GPIO	General-purpose I/O	

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
ARMCOP	R	0	0	0	0	0	0	0	0
	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

= Unimplemented or Reserved

Figure 4-3. CRG Register Summary (continued)

4.3.2.1 CRG Synthesizer Register (SYNR)

The SYNR register controls the multiplication factor of the PLL. If the PLL is on, the count in the loop divider (SYNR) register effectively multiplies up the PLL clock (PLLCLK) from the reference frequency by 2 x (SYNR+1). PLLCLK will not be below the minimum VCO frequency (f_{SCM}).

$$PLLCLK = 2 \times OSCCLK \times \frac{(SYNR + 1)}{(REFDV + 1)}$$

NOTE

If PLL is selected (PLLSEL=1), Bus Clock = PLLCLK / 2
Bus Clock must not exceed the maximum operating system frequency.

	7	6	5	4	3	2	1	0
R	0	0	SYN5	SYNR	SYN3	SYN2	SYN1	SYN0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 4-4. CRG Synthesizer Register (SYNR)

Read: anytime

Write: anytime except if PLLSEL = 1

NOTE

Write to this register initializes the lock detector bit and the track detector bit.

8.1.4 Block Diagram

Figure 8-1 is a high level block diagram of the SCI module, showing the interaction of various function blocks.

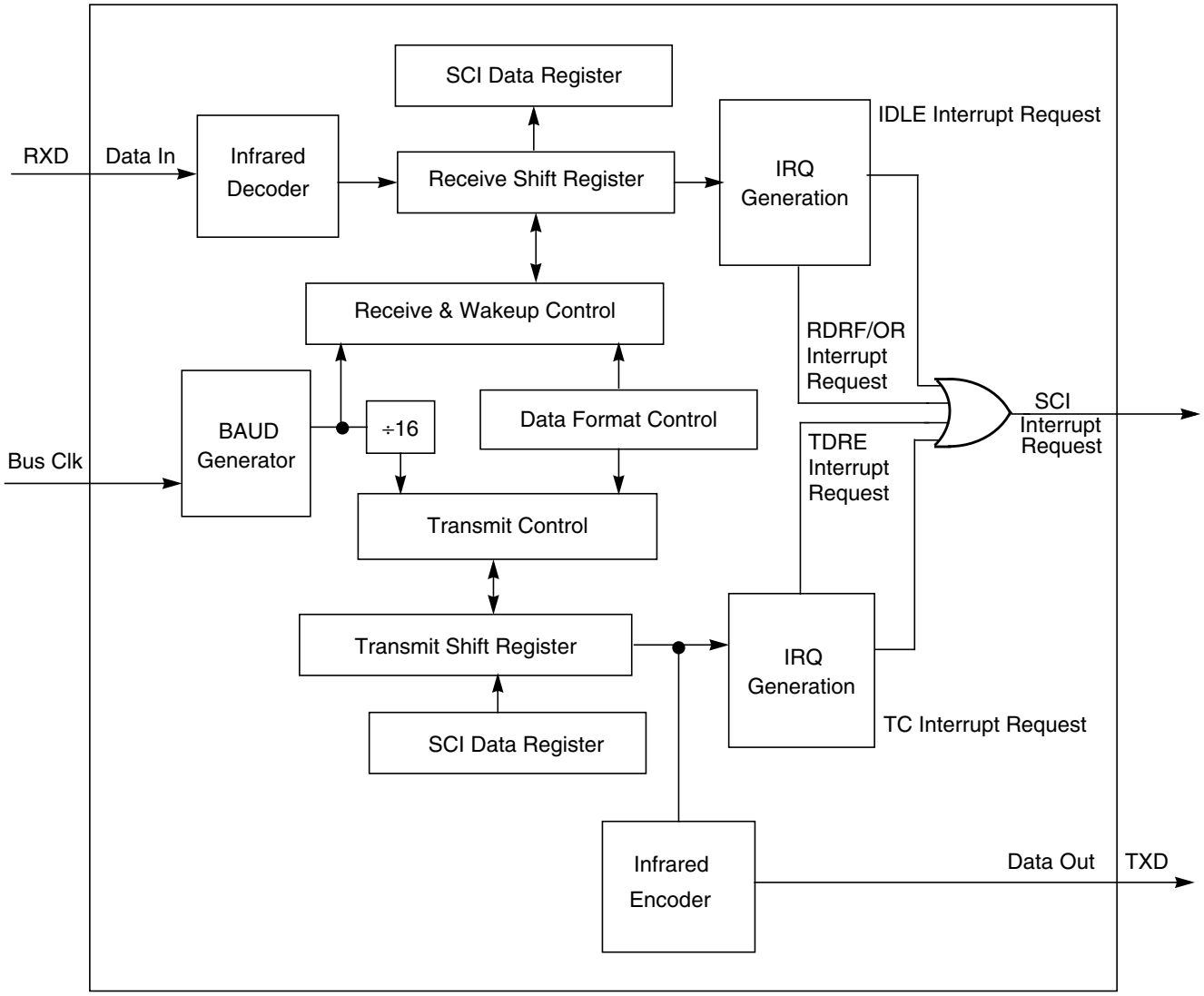


Figure 8-1. SCI Block Diagram

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
SCICR2	R	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
	W								
SCISR1	R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
	W								
SCISR2	R	0	0	0	0	0	BRK13	TXDIR	RAF
	W								
SCIDRH	R	R8	T8	0	0	0	0	0	0
	W								
SCIDRL	R	R7	R6	R5	R4	R3	R2	R1	R0
	W	T7	T6	T5	T4	T3	T2	T1	T0

= Unimplemented or Reserved

Figure 8-2. SCI Registers Summary

In Figure 8-16 the verification samples RT3 and RT5 determine that the first low detected was noise and not the beginning of a start bit. The RT clock is reset and the start bit search begins again. The noise flag is not set because the noise occurred before the start bit was found.

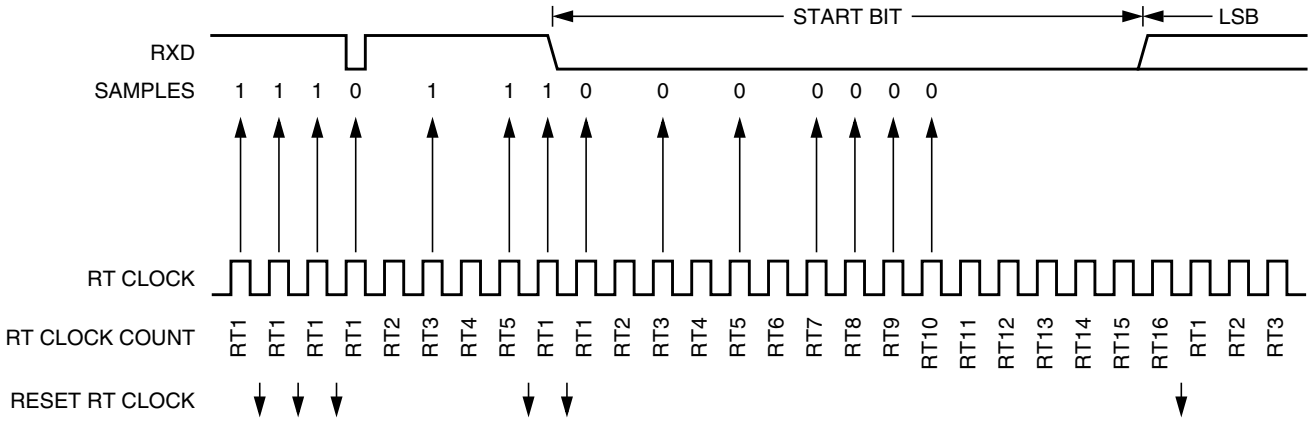


Figure 8-16. Start Bit Search Example 1

In Figure 8-17, verification sample at RT3 is high. The RT3 sample sets the noise flag. Although the perceived bit time is misaligned, the data samples RT8, RT9, and RT10 are within the bit time and data recovery is successful.

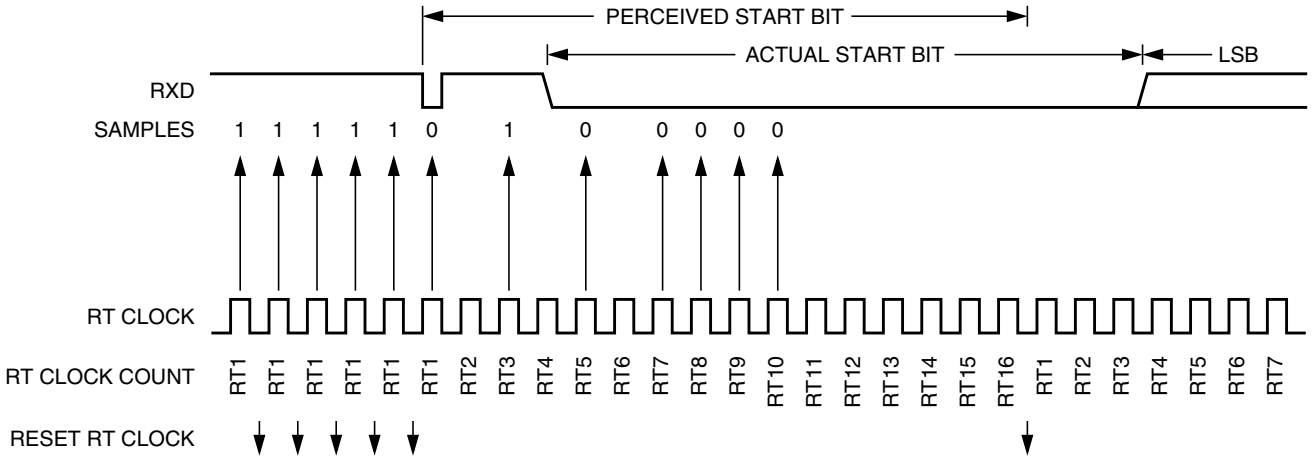


Figure 8-17. Start Bit Search Example 2

Table 10-5. IIC Divider and Hold Values (Sheet 2 of 5)

IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
23	256	33	126	129
24	288	49	142	145
25	320	49	158	161
26	384	65	190	193
27	480	65	238	241
28	320	33	158	161
29	384	33	190	193
2A	448	65	222	225
2B	512	65	254	257
2C	576	97	286	289
2D	640	97	318	321
2E	768	129	382	385
2F	960	129	478	481
30	640	65	318	321
31	768	65	382	385
32	896	129	446	449
33	1024	129	510	513
34	1152	193	574	577
35	1280	193	638	641
36	1536	257	766	769
37	1920	257	958	961
38	1280	129	638	641
39	1536	129	766	769
3A	1792	257	894	897
3B	2048	257	1022	1025
3C	2304	385	1150	1153
3D	2560	385	1278	1281
3E	3072	513	1534	1537
3F	3840	513	1918	1921
MUL=2				
40	40	14	12	22
41	44	14	14	24
42	48	16	16	26
43	52	16	18	28
44	56	18	20	30
45	60	18	22	32
46	68	20	26	36
47	80	20	32	42
48	56	14	20	30
49	64	14	24	34
4A	72	18	28	38
4B	80	18	32	42
4C	88	22	36	46
4D	96	22	40	50
4E	112	26	48	58

Table 10-5. IIC Divider and Hold Values (Sheet 5 of 5)

IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
A8	1280	132	632	644
A9	1536	132	760	772
AA	1792	260	888	900
AB	2048	260	1016	1028
AC	2304	388	1144	1156
AD	2560	388	1272	1284
AE	3072	516	1528	1540
AF	3840	516	1912	1924
B0	2560	260	1272	1284
B1	3072	260	1528	1540
B2	3584	516	1784	1796
B3	4096	516	2040	2052
B4	4608	772	2296	2308
B5	5120	772	2552	2564
B6	6144	1028	3064	3076
B7	7680	1028	3832	3844
B8	5120	516	2552	2564
B9	6144	516	3064	3076
BA	7168	1028	3576	3588
BB	8192	1028	4088	4100
BC	9216	1540	4600	4612
BD	10240	1540	5112	5124
BE	12288	2052	6136	6148
BF	15360	2052	7672	7684

10.3.2.3 IIC Control Register (IBCR)

	7	6	5	4	3	2	1	0
R	IBEN	IBIE	MS/SL	Tx/Rx	TXAK	0	0	IBSWAI
W						RSTA		
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 10-6. IIC Bus Control Register (IBCR)

Read and write anytime

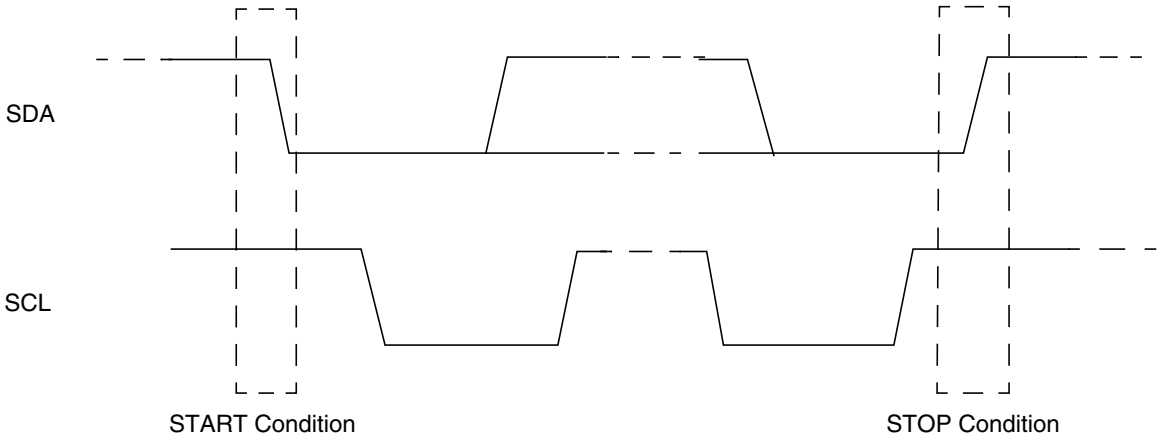


Figure 10-10. Start and Stop Conditions

10.4.1.2 Slave Address Transmission

The first byte of data transfer immediately after the START signal is the slave address transmitted by the master. This is a seven-bit calling address followed by a R/W bit. The R/W bit tells the slave the desired direction of data transfer.

1 = Read transfer, the slave transmits data to the master.
0 = Write transfer, the master transmits data to the slave.

Only the slave with a calling address that matches the one transmitted by the master will respond by sending back an acknowledge bit. This is done by pulling the SDA low at the 9th clock (see Figure 10-9).

No two slaves in the system may have the same address. If the IIC bus is master, it must not transmit an address that is equal to its own slave address. The IIC bus cannot be master and slave at the same time.However, if arbitration is lost during an address cycle the IIC bus will revert to slave mode and operate correctly even if it is being addressed by another master.

10.4.1.3 Data Transfer

As soon as successful slave addressing is achieved, the data transfer can proceed byte-by-byte in a direction specified by the R/W bit sent by the calling master

All transfers that come after an address cycle are referred to as data transfers, even if they carry sub-address information for the slave device.

Each data byte is 8 bits long. Data may be changed only while SCL is low and must be held stable while SCL is high as shown in Figure 10-9. There is one clock pulse on SCL for each data bit, the MSB being transferred first. Each data byte has to be followed by an acknowledge bit, which is signalled from the receiving device by pulling the SDA low at the ninth clock. So one complete data byte transfer needs nine clock pulses.

If the slave receiver does not acknowledge the master, the SDA line must be left high by the slave. The master can then generate a stop signal to abort the data transfer or a start signal (repeated start) to commence a new calling.

Table 11-5. MII Management Frame Operation

BUSY	OP	Operation
1	xx	Ignore
0	00	Ignore
0	01	Write
0	10	Read
0	11	Ignore

BUSY — Operation in Progress

This read-only status bit indicates MII management activity. BUSY is asserted after a valid OP write and is cleared when the MMCIF flag is set.

1 = MII is busy (operation in progress).
0 = MII is idle (ready for operation).

NOPRE — No Preamble

Any value written while BUSY is set is ignored. The IEEE 802.3 standard allows the preamble to be dropped if the attached PHY does not require it. While this bit is set, a preamble is not prepended to the MII management frame.

1 = No preamble is sent.
0 = 32-bit preamble is sent.

MDCSEL — Management Clock Rate Select

Any value programmed while BUSY bit is set is ignored. This field controls the frequency of the MII management data clock (MDC) relative to the IP bus clock. MDC toggles only during a valid MII management transaction. While MDC is not active, it remains low. Any nonzero value results in an MDC frequency given by the following formula:

MDC frequency = Bus clock frequency / (2 * MDCSEL)

The MDCSEL field must be programmed with a value to provide an MDC frequency of less-than or equal-to 2.5 MHz to be compliant with the IEEE MII specification. The MDCSEL must be set to a nonzero value in order to source a read or write MII management frame.

Table 11-6. Programming Examples for MDCSEL

IP Bus Clock Frequency	MDCSEL	MDC Frequency
20 MHz	0x4	2.5 MHz
25 MHz	0x5	2.5 MHz
33 MHz	0x7	2.36 MHz
40 MHz	0x8	2.5 MHz
50 MHz	0xA	2.5 MHz

Table 11-8. Miscellaneous Fields

Index	Field	Unit
0-2	None	Read 0s
3	TXBYT	Bytes
4	BSLOT	Slot time
5	RETX	Retransmissions
6	RANDOM	N/A
7	Reserved	Reserved

TXBYT — Transmit Frame Byte Counter

This11-bit read-only field indicates the number of bytes of the current frame that have been read from the transmit buffer by the EMAC transmitter. This register does not include transmitted pad data that is added to frames if less than the minimum amount of data is transmitted nor the FCS data that is appended to the end of transmit frames. While sending pause frames with the PAUSE command, this register is ignored.

BSLOT — Backoff Slot Time Counter

This 10-bit read-only field indicates the number of slot times (512 bit times) in progress during the backoff delay. This counter clears at the end of backoff delay, which is set by the random algorithm. The MISC[10] bit reads 0.

RETX — Retransmission Counter

This 4-bit read-only field indicates the current retransmission count if retransmission takes place due to collision. The MISC[10:4] bits read 0.

RANDOM — Backoff Random Number

This10-bit read-only random number is generated for use by the backoff logic. The value returned when reading this field is random if the transmitter is enabled. The MISC[10] bit reads 0.

11.4 Functional Description

The EMAC provides a 10/100 Mbps Ethernet media access control (MAC) function and is designed to connect to a PHY device supporting MII. The EMAC is an 802.3 compliant Ethernet controller specifically optimized for 8-/16-bit embedded processors. The main components of the EMAC are the receiver, transmitter, MAC flow control, MII management, and receive and transmit Ethernet buffer interfaces.

11.4.1 Ethernet Frame

In an Ethernet network, information is received or transmitted in the form of a frame. The frame format used for Ethernet consists of preamble (PA), start frame delimiter (SFD), destination address (DA), source address (SA), type/length field, data field, and frame check sequence (FCS). See Table 11-9.

Table 11-9. Ethernet Frame Structure

Preamble	Start Frame Delimiter	Destination Address	Source Address	Type/ Length	Data	Frame Check Sequence
7 bytes	1 bytes	6 bytes	6 bytes	2 bytes	46 to 1500 bytes	4 bytes

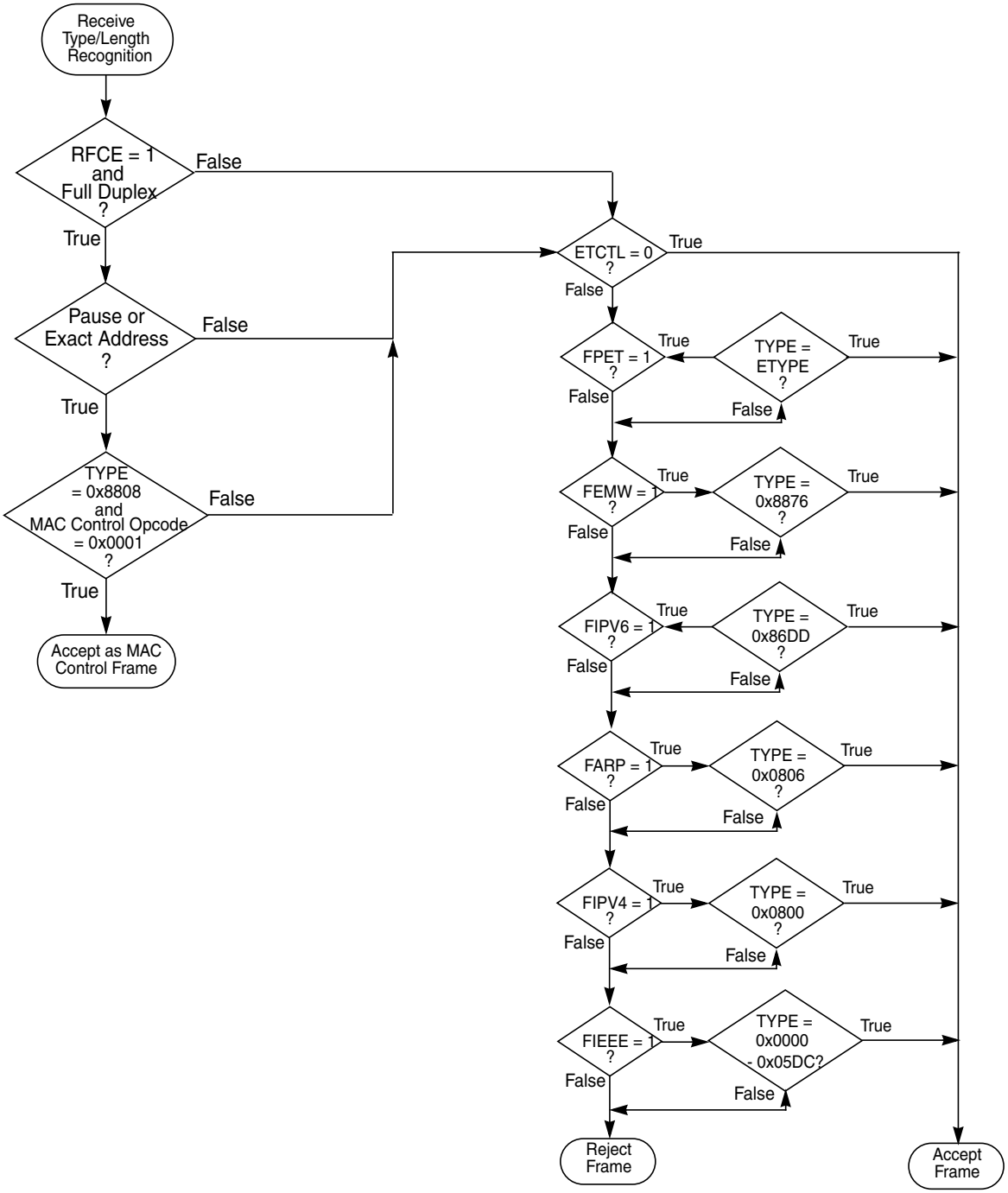


Figure 11-25. Receive Type/Length Recognition Algorithm

11.4.2.2.1 Ethertype Filter

While any of the ETCTL register bits are set, the Ethertype filter is enabled to reject frames that are not standard Ethernet protocols. In this case, the collection of set bits determines which Ethertypes are accepted; all other Ethertypes are rejected. If all bits of the ETCTL register are clear, Ethertype filtering is

FEFLTD — Far End Fault Disable
1 = Far end fault detect is disabled
0 = Far end fault detect on receive and transmit is enabled. This applies only while auto-negotiation is disabled

MIILBO — MII Loopback Disable
1 = Disable MII loopback
0 = MII transmit data is looped back to the MII receive pins

JBDE — Jabber Detect Enable (10BASE-T)
1 = Enable jabber detection
0 = Disable jabber detection

LNKTSTD — Link Test Disable (10BASE-T)
1 = Disable 10BASE-T link integrity test
0 = 10BASE-T link integrity test enabled

POLCORD — Disable Polarity Correction (10BASE-T)
1 = 10BASE-T receive polarity correction is disabled
0 = 10BASE-T receive polarity is automatically corrected

ALGD — Disable Alignment
1 = Un-aligned mode. Available only in symbol mode
0 = Aligned mode

ENCBYP — Encoder Bypass
1 = Symbol mode and bypass 4B/5B encoder and decoder
0 = Normal mode

SCRBYP — Scrambler Bypass Mode (100BASE-TX)
1 = Bypass the scrambler and de-scrambler
0 = Normal

TRDANALB — Transmit and Receive Disconnect and Analog Loopback
1 = High-impedance twisted pair transmitter. Analog loopback mode overrides and forces this bit
0 = Normal operation

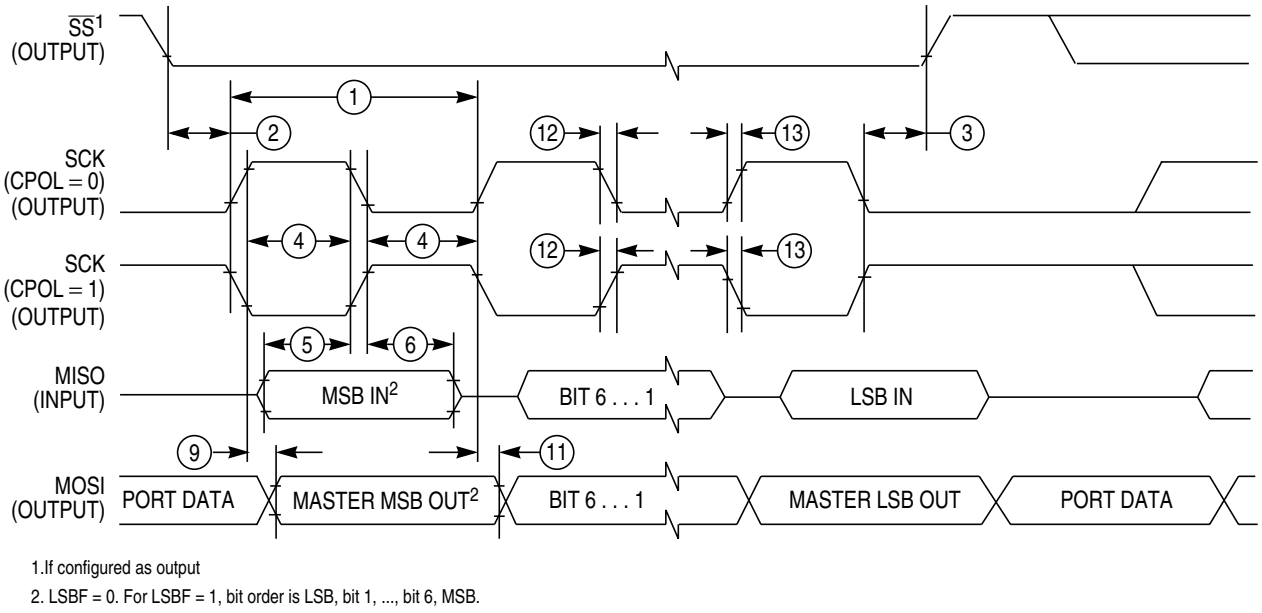
TRTST — Transmit and Receive Test (100BASE-TX)
1 = Transmit and receive data regardless of link status
0 = Normal operation

12.4 Functional Description

The EPHY is an IEEE 802.3 compliant 10/100 Ethernet physical transceiver. The EPHY can be configured to support 10BASE-T or 100BASE-TX applications. The EPHY is configurable via internal registers which are accessible through the MII management interface as well as limited configurability using the EPHY register map.

There are five basic modes of operation for the EPHY:

- Power down/initialization
- Auto-negotiate



1.If configured as output

2. LSBF = 0. For LSBF = 1, bit order is LSB, bit 1, ..., bit 6, MSB.

Figure A-12. SPI Master Timing (CPHA=1)

In Table A-28 the timing characteristics for master mode are listed.

Table A-28. SPI Master Mode Timing Characteristics

Num	C	Characteristic	Symbol	Min	Typ	Max	Unit
1	P	SCK Frequency	f_{sck}	1/2048	—	1/2	f_{bus}
1	P	SCK Period	t_{sck}	2	—	2048	t_{bus}
2	D	Enable Lead Time	t_{lead}	—	1/2	—	t_{sck}
3	D	Enable Lag Time	t_{lag}	—	1/2	—	t_{sck}
4	D	Clock (SCK) High or Low Time	t_{wsck}	—	1/2	—	t_{sck}
5	D	Data Setup Time (Inputs)	t_{su}	8	—	—	ns
6	D	Data Hold Time (Inputs)	t_{hi}	8	—	—	ns
9	D	Data Valid after SCK Edge	t_{vsck}	—	—	30	ns
10	D	Data Valid after $\overline{SS}$ fall (CPHA=0)	t_{vss}	—	—	15	ns
11	D	Data Hold Time (Outputs)	t_{ho}	20	—	—	ns
12	D	Rise and Fall Time Inputs	t_{rfi}	—	—	8	ns
13	D	Rise and Fall Time Outputs	t_{rfo}	—	—	8	ns

A.16.2 Slave Mode

In Figure A-13 the timing diagram for slave mode with transmission format CPHA = 0 is depicted.