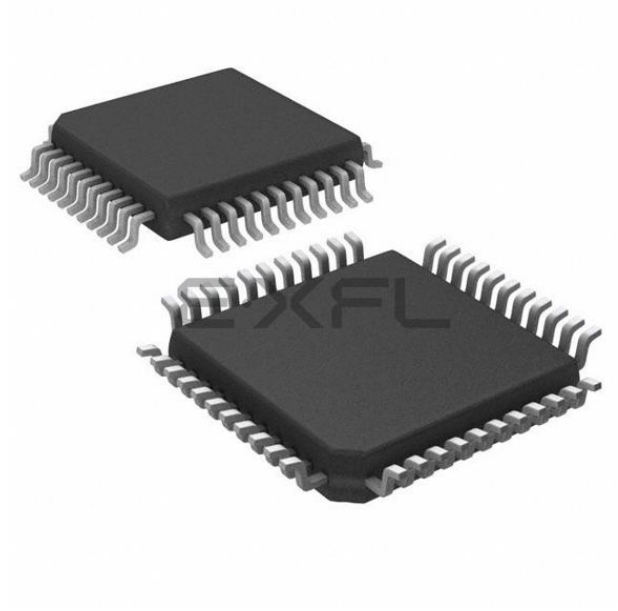


Welcome to E-XFL.COM

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	HC11
Core Size	8-Bit
Speed	2MHz
Connectivity	SCI, SPI
Peripherals	POR, WDT
Number of I/O	26
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	192 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-QFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68l11d0cfbe2

Chapter 5
Input/Output (I/O) Ports

Chapter 6
Serial Communications Interface (SCI)

Chapter 7
Serial Peripheral Interface (SPI)

Chapter 8
Programmable Timer

Chapter 9
Electrical Characteristics

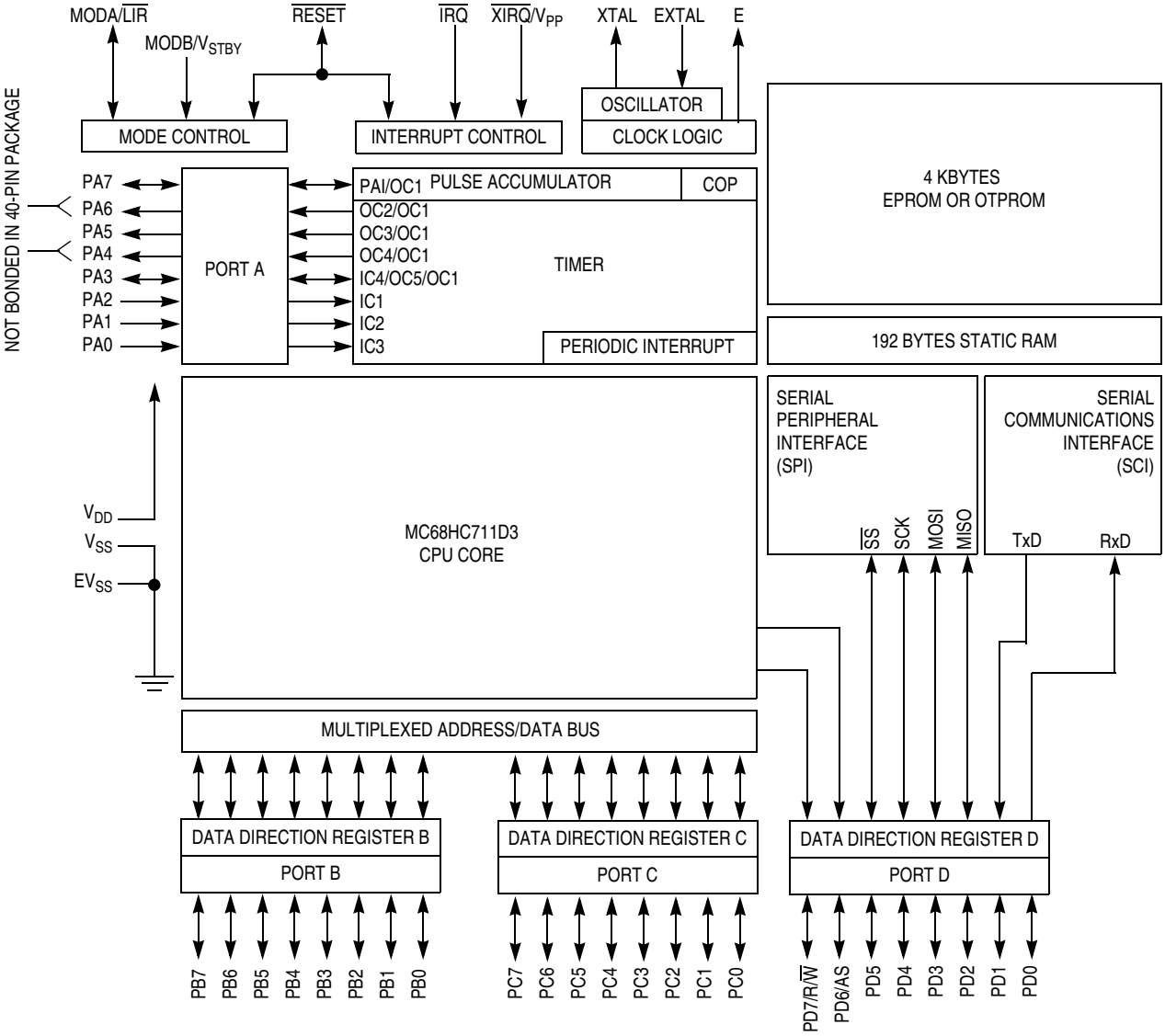


Figure 1-1. MC68HC711D3 Block Diagram

1.4 Pin Descriptions

Refer to Figure 1-2, Figure 1-3, and Figure 1-4 for pin assignments.

2.4.2 Programming the EPROM with Downloaded Data

When using this method, the EPROM is programmed by software while in the special test or bootstrap modes. User-developed software can be uploaded through the SCI or a ROM-resident EPROM programming utility can be used. The 12-volt nominal programming voltage must be present on the $\overline{\text{XIRQ}}/V_{PP}$ pin. To use the resident utility, bootload a 3-byte program consisting of a single jump instruction to $BF00. $BF00 is the starting address of a resident EPROM programming utility. The utility program sets the X and Y index registers to default values, then receives programming data from an external host, and puts it in EPROM. The value in IX determines programming delay time. The value in IY is a pointer to the first address in EPROM to be programmed (default = $F000).

When the utility program is ready to receive programming data, it sends the host the $FF character. Then it waits. When the host sees the $FF character, the EPROM programming data is sent, starting with the first location in the EPROM array. After the last byte to be programmed is sent and the corresponding verification data is returned, the programming operation is terminated by resetting the MCU.

2.4.3 PROM Programming Control Register

The PROM programming control register (PPROG) is used to control the programming of the OTPROM or EPROM. PPROG is cleared on reset so that the PROM is configured for normal read.

Address: $003B

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	MBE	0	ELAT	EXCOL	EXROW	0	0	PGM
Reset:	0	0	0	0	0	0	0	0

Figure 2-5. PROM Programming Control Register (PPROG)

MBE — Multiple Byte Program Enable Bit
This bit is reserved for testing.

Bit 6, 2, and 1 — Not implemented
Always read 0.

ELAT — EPROM (OTPROM) Latch Control Bit
1 = PROM address and data bus are configured for programming. Writes to PROM cause address and data to be latched. The PROM cannot be read.
0 = PROM address and data bus are configured for normal reads. PROM cannot be programmed.

EXCOL — Select Extra Columns Bit
This bit is reserved for testing.

EXROW — Select Extra Row Bit
This bit is reserved for testing.

PGM — EPROM (OTPROM) Program Command Bit
This bit may be written only when ELAT = 1.
1 = Programming power is switched on to PROM array.
0 = Programming power is switched off.

A byte is eight bits wide and can be accessed at any byte location. A word is composed of two consecutive bytes with the most significant byte at the lower value address. Because the M68HC11 is an 8-bit CPU, there are no special requirements for alignment of instructions or operands.

3.4 Opcodes and Operands

The M68HC11 Family of microcontrollers uses 8-bit opcodes. Each opcode identifies a particular instruction and associated addressing mode to the CPU. Several opcodes are required to provide each instruction with a range of addressing capabilities. Only 256 opcodes would be available if the range of values were restricted to the number able to be expressed in 8-bit binary numbers.

A 4-page opcode map has been implemented to expand the number of instructions. An additional byte, called a prebyte, directs the processor from page 0 of the opcode map to one of the other three pages. As its name implies, the additional byte precedes the opcode.

A complete instruction consists of a prebyte, if any, an opcode, and zero, one, two, or three operands. The operands contain information the CPU needs for executing the instruction. Complete instructions can be from one to five bytes long.

3.5 Addressing Modes

Six addressing modes can be used to access memory:

1. Immediate
2. Direct
3. Extended
4. Indexed
5. Inherent
6. Relative

These modes are detailed in the following paragraphs. All modes except inherent mode use an effective address. The effective address is the memory address from which the argument is fetched or stored or the address from which execution is to proceed. The effective address can be specified within an instruction, or it can be calculated.

3.5.1 Immediate

In the immediate addressing mode, an argument is contained in the byte(s) immediately following the opcode. The number of bytes following the opcode matches the size of the register or memory location being operated on. There are 2-, 3-, and 4- (if prebyte is required) byte immediate instructions. The effective address is the address of the byte following the instruction.

3.5.2 Direct

In the direct addressing mode, the low-order byte of the operand address is contained in a single byte following the opcode, and the high-order byte of the address is assumed to be $00. Addresses $00–$FF are thus accessed directly, using 2-byte instructions. Execution time is reduced by eliminating the additional memory access required for the high-order address byte. In most applications, this 256-byte area is reserved for frequently referenced data. In M68HC11 MCUs, the memory map can be configured for combinations of internal registers, RAM, or external memory to occupy these addresses.

Table 3-2. Instruction Set (Sheet 3 of 8)

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
BITB (opr)	Bit(s) Test B with Memory	B • M	B IMM B DIR B EXT B IND,X B IND,Y	C5 D5 F5 E5 18 E5	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
BLE (rel)	Branch if Δ Zero	? Z + (N ⊕ V) = 1	REL	2F	rr	3	—	—	—	—	—	—	—	—
BLO (rel)	Branch if Lower	? C = 1	REL	25	rr	3	—	—	—	—	—	—	—	—
BLS (rel)	Branch if Lower or Same	? C + Z = 1	REL	23	rr	3	—	—	—	—	—	—	—	—
BLT (rel)	Branch if < Zero	? N ⊕ V = 1	REL	2D	rr	3	—	—	—	—	—	—	—	—
BMI (rel)	Branch if Minus	? N = 1	REL	2B	rr	3	—	—	—	—	—	—	—	—
BNE (rel)	Branch if not = Zero	? Z = 0	REL	26	rr	3	—	—	—	—	—	—	—	—
BPL (rel)	Branch if Plus	? N = 0	REL	2A	rr	3	—	—	—	—	—	—	—	—
BRA (rel)	Branch Always	? 1 = 1	REL	20	rr	3	—	—	—	—	—	—	—	—
BRCLR(opr) (msk) (rel)	Branch if Bit(s) Clear	? M • mm = 0	DIR IND,X IND,Y	13 1F 18 1F	dd mm rr ff mm rr ff mm rr	6 7 8	—	—	—	—	—	—	—	—
BRN (rel)	Branch Never	? 1 = 0	REL	21	rr	3	—	—	—	—	—	—	—	—
BRSET(opr) (msk) (rel)	Branch if Bit(s) Set	? (M̄) • mm = 0	DIR IND,X IND,Y	12 1E 18 1E	dd mm rr ff mm rr ff mm rr	6 7 8	—	—	—	—	—	—	—	—
BSET (opr) (msk)	Set Bit(s)	M + mm ⇒ M	DIR IND,X IND,Y	14 1C 18 1C	dd mm ff mm ff mm	6 7 8	—	—	—	—	Δ	Δ	0	—
BSR (rel)	Branch to Subroutine	See Figure 3-2	REL	8D	rr	6	—	—	—	—	—	—	—	—
BVC (rel)	Branch if Overflow Clear	? V = 0	REL	28	rr	3	—	—	—	—	—	—	—	—
BVS (rel)	Branch if Overflow Set	? V = 1	REL	29	rr	3	—	—	—	—	—	—	—	—
CBA	Compare A to B	A – B	INH	11	—	2	—	—	—	—	Δ	Δ	Δ	Δ
CLC	Clear Carry Bit	0 ⇒ C	INH	0C	—	2	—	—	—	—	—	—	—	0
CLI	Clear Interrupt Mask	0 ⇒ I	INH	0E	—	2	—	—	—	0	—	—	—	—
CLR (opr)	Clear Memory Byte	0 ⇒ M	EXT IND,X IND,Y	7F 6F 18 6F	hh ll ff ff	6 6 7	—	—	—	—	0	1	0	0
CLRA	Clear Accumulator A	0 ⇒ A	A INH	4F	—	2	—	—	—	—	0	1	0	0
CLRB	Clear Accumulator B	0 ⇒ B	B INH	5F	—	2	—	—	—	—	0	1	0	0
CLV	Clear Overflow Flag	0 ⇒ V	INH	0A	—	2	—	—	—	—	—	—	0	—
CMPA (opr)	Compare A to Memory	A – M	A IMM A DIR A EXT A IND,X A IND,Y	81 91 B1 A1 18 A1	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
CMPB (opr)	Compare B to Memory	B – M	B IMM B DIR B EXT B IND,X B IND,Y	C1 D1 F1 E1 18 E1	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
COM (opr)	Ones Complement Memory Byte	$FF – M ⇒ M	EXT IND,X IND,Y	73 63 18 63	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	0	1

Table 3-2. Instruction Set (Sheet 7 of 8)

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
RTS	Return from Subroutine	See Figure 3-2	INH	39	—	5	—	—	—	—	—	—	—	—
SBA	Subtract B from A	A – B ⇒ A	INH	10	—	2	—	—	—	—	Δ	Δ	Δ	Δ
SBCA (opr)	Subtract with Carry from A	A – M – C ⇒ A	A IMM A DIR A EXT A IND,X A IND,Y	82 92 B2 A2 18 A2	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
SBCB (opr)	Subtract with Carry from B	B – M – C ⇒ B	B IMM B DIR B EXT B IND,X B IND,Y	C2 D2 F2 E2 18 E2	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
SEC	Set Carry	1 ⇒ C	INH	0D	—	2	—	—	—	—	—	—	—	1
SEI	Set Interrupt Mask	1 ⇒ I	INH	0F	—	2	—	—	—	1	—	—	—	—
SEV	Set Overflow Flag	1 ⇒ V	INH	0B	—	2	—	—	—	—	—	—	1	—
STAA (opr)	Store Accumulator A	A ⇒ M	A DIR A EXT A IND,X A IND,Y	97 B7 A7 18 A7	dd hh ll ff ff	3 4 4 5	—	—	—	—	Δ	Δ	0	—
STAB (opr)	Store Accumulator B	B ⇒ M	B DIR B EXT B IND,X B IND,Y	D7 F7 E7 18 E7	dd hh ll ff ff	3 4 4 5	—	—	—	—	Δ	Δ	0	—
STD (opr)	Store Accumulator D	A ⇒ M, B ⇒ M + 1	DIR EXT IND,X IND,Y	DD FD ED 18 ED	dd hh ll ff ff	4 5 5 6	—	—	—	—	Δ	Δ	0	—
STOP	Stop Internal Clocks	—	INH	CF	—	2	—	—	—	—	—	—	—	—
STS (opr)	Store Stack Pointer	SP ⇒ M : M + 1	DIR EXT IND,X IND,Y	9F BF AF 18 AF	dd hh ll ff ff	4 5 5 6	—	—	—	—	Δ	Δ	0	—
STX (opr)	Store Index Register X	IX ⇒ M : M + 1	DIR EXT IND,X IND,Y	DF FF EF CD EF	dd hh ll ff ff	4 5 5 6	—	—	—	—	Δ	Δ	0	—
STY (opr)	Store Index Register Y	IY ⇒ M : M + 1	DIR EXT IND,X IND,Y	18 DF 18 FF 1A EF 18 EF	dd hh ll ff ff	5 6 6 6	—	—	—	—	Δ	Δ	0	—
SUBA (opr)	Subtract Memory from A	A – M ⇒ A	A IMM A DIR A EXT A IND,X A IND,Y	80 90 B0 A0 18 A0	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
SUBB (opr)	Subtract Memory from B	B – M ⇒ B	A IMM A DIR A EXT A IND,X A IND,Y	C0 D0 F0 E0 18 E0	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
SUBD (opr)	Subtract Memory from D	D – M : M + 1 ⇒ D	IMM DIR EXT IND,X IND,Y	83 93 B3 A3 18 A3	jj kk dd hh ll ff ff	4 5 6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
SWI	Software Interrupt	See Figure 3-2	INH	3F	—	14	—	—	—	1	—	—	—	—
TAB	Transfer A to B	A ⇒ B	INH	16	—	2	—	—	—	—	Δ	Δ	0	—
TAP	Transfer A to CC Register	A ⇒ CCR	INH	06	—	2	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ

Table 3-2. Instruction Set (Sheet 8 of 8)

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
TBA	Transfer B to A	B ⇒ A	INH	17	—	2	—	—	—	—	Δ	Δ	0	—
TEST	TEST (Only in Test Modes)	Address Bus Counts	INH	00	—	*	—	—	—	—	—	—	—	—
TPA	Transfer CC Register to A	CCR ⇒ A	INH	07	—	2	—	—	—	—	—	—	—	—
TST (opr)	Test for Zero or Minus	M – 0	EXT IND,X IND,Y	7D 6D 18 6D	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	0	0
TSTA	Test A for Zero or Minus	A – 0	A INH	4D	—	2	—	—	—	—	Δ	Δ	0	0
TSTB	Test B for Zero or Minus	B – 0	B INH	5D	—	2	—	—	—	—	Δ	Δ	0	0
TSX	Transfer Stack Pointer to X	SP + 1 ⇒ IX	INH	30	—	3	—	—	—	—	—	—	—	—
TSY	Transfer Stack Pointer to Y	SP + 1 ⇒ IY	INH	18 30	—	4	—	—	—	—	—	—	—	—
TXS	Transfer X to Stack Pointer	IX – 1 ⇒ SP	INH	35	—	3	—	—	—	—	—	—	—	—
TYS	Transfer Y to Stack Pointer	IY – 1 ⇒ SP	INH	18 35	—	4	—	—	—	—	—	—	—	—
WAI	Wait for Interrupt	Stack Regs & WAIT	INH	3E	—	**	—	—	—	—	—	—	—	—
XGDX	Exchange D with X	IX ⇒ D, D ⇒ IX	INH	8F	—	3	—	—	—	—	—	—	—	—
XGDY	Exchange D with Y	IY ⇒ D, D ⇒ IY	INH	18 8F	—	4	—	—	—	—	—	—	—	—

Cycle

* Infinity or until reset occurs

** 12 cycles are used beginning with the opcode fetch. A wait state is entered which remains in effect for an integer number of MPU E-clock cycles (n) until an interrupt is recognized. Finally, two additional cycles are used to fetch the appropriate interrupt vector (14 + n total).

Operands

dd = 8-bit direct address ($0000–$00FF) (high byte assumed to be $00)
ff = 8-bit positive offset $00 (0) to $FF (255) (is added to index)
hh = High-order byte of 16-bit extended address
ii = One byte of immediate data
jj = High-order byte of 16-bit immediate data
kk = Low-order byte of 16-bit immediate data
ll = Low-order byte of 16-bit extended address
mm = 8-bit mask (set bits to be affected)
rr = Signed relative offset $80 (–128) to $7F (+127)
(offset relative to address following machine code offset byte)

Operators

( ) Contents of register shown inside parentheses
⇐ Is transferred to
⇑ Is pulled from stack
⇓ Is pushed onto stack
• Boolean AND
\+ Arithmetic addition symbol except where used as inclusive-OR symbol in Boolean formula
⊕ Exclusive-OR
* Multiply
: Concatenation
– Arithmetic subtraction symbol or negation symbol (two's complement)

Condition Codes

— Bit not changed
0 Bit always cleared
1 Bit always set
Δ Bit cleared or set, depending on operation
↓ Bit can be cleared, cannot become set

4.3.6 Highest Priority I Interrupt and Miscellaneous Register (HPRIO)

Four bits of this register (PSEL3–PSEL0) are used to select one of the I bit related interrupt sources and to elevate it to the highest I bit masked position of the priority resolution circuit. In addition, four miscellaneous system control bits are included in this register.

Address: $003C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	RBOOT	SMOD	MDA	IRVNE	PSEL3	PSEL2	PSEL1	PSEL0
Write:								
Reset:	Note 1				0	1	0	1

1. The values of the RBOOT, SMOD, IRVNE, and MDA bits at reset depend on the mode during initialization. Refer to Table 4-3.

Figure 4-7. Highest Priority I-Bit Interrupt and Miscellaneous Register (HPRIO)

RBOOT — Read Bootstrap ROM

This bit can be read at any time. It can be written only in special modes (SMOD = 1). In special bootstrap mode, it is set during reset. Reset clears it in all other modes.

1 = Bootloader ROM is enabled in the memory map at $BF00–$BFFF.
0 = Bootloader ROM is disabled and is not in the memory map.

SMOD and MDA — Special Mode Select and Mode Select A

These two bits can be read at any time.These bits reflect the status of the MODA and MODB input pins at the rising edge of reset. SMOD may be written only in special modes. It cannot be written to a 1 after being cleared without an interim reset. MDA may be written at any time in special modes, but only once in normal modes. An interpretation of the values of these two bits is shown in Table 4-3.

Table 4-3. Hardware Mode Select Summary

Inputs		Mode	Latched at Reset	
MODB	MODA		SMOD	MDA
1	0	Single chip	0	0
1	1	Expanded multiplexed	0	1
0	0	Special bootstrap	1	0
0	1	Special test	1	1

IRVNE — Internal Read Visibility/Not E

This bit may be read at any time. It may be written once in any mode. IRVNE is set during reset in special test mode only, and cleared by reset in the other modes.

1 = Data from internal reads is driven out on the external data bus in expanded modes.
0 = Data from internal reads is not visible on the external data bus.

As shown in the table, in single-chip and bootstrap modes IRVNE determines whether the E clock is driven out or forced low.

1 = E pin driven low
0 = E clock driven out of the chip

4.4 Low-Power Operation

The M68HC11 Family of microcontroller units (MCU) has two programmable low power-consumption modes: stop and wait. In the wait mode, the on-chip oscillator remains active. In the stop mode, the oscillator is stopped. This subsection describes these two low power-consumption modes.

4.4.1 Stop Mode

The STOP instruction places the MCU in its lowest power-consumption mode, provided the S bit in the CCR is cleared. In this mode, all clocks are stopped, thereby halting all internal processing.

To exit the stop mode, a low level must be applied to either the $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$, or $\overline{\text{RESET}}$ pin. An external interrupt used at $\overline{\text{IRQ}}$ is only effective if the I bit in the CCR is cleared. An external interrupt applied at the $\overline{\text{XIRQ}}$ input is effective, regardless of the setting of the X bit of the CCR. However, the actual recovery sequence differs, depending on the X bit setting. If the X bit is cleared, the MCU starts with the stacking sequence leading to the normal service of the $\overline{\text{XIRQ}}$ request. If the X bit is set, the processing always continues with the instruction immediately following the STOP instruction. A low input to the $\overline{\text{RESET}}$ pin always results in an exit from the stop mode, and the start of MCU operations is determined by the reset vector.

The CPU will not exit stop mode correctly when interrupted by $\overline{\text{IRQ}}$ or $\overline{\text{XIRQ}}$ if the instruction preceding STOP is a column 4 or 5 accumulator inherent (opcodes $4X and $5X) instruction, such as NEGA, NEGB, COMA, COMB, etc. These single-byte, two-cycle instructions must be followed by an NOP, then the STOP command. If reset is used to exit stop mode, the CPU will respond properly.

A restart delay is required if the internal oscillator is being used. The delay allows the oscillator to stabilize when exiting the stop mode. If a stable external oscillator is being used, the delay (DLY) bit in the OPTION register can be cleared to bypass the delay. If the DLY bit is clear, the $\overline{\text{RESET}}$ pin would not normally be used to exit the stop mode. The reset sequence sets the DLY bit, and the restart delay would be reimposed.

4.4.2 Wait Mode

The wait (WAI) instruction places the MCU in a low power-consumption mode. The wait mode consumes more power than the stop mode since the oscillator is kept running. Upon execution of the WAI instruction, the machine state is stacked and program execution stops.

The wait state can be exited only by an unmasked interrupt or $\overline{\text{RESET}}$. If the I bit of the CCR is set and the COP is disabled, the timer system is turned off by WAI to further reduce power consumption. The amount of power savings is application dependent. It also depends upon the circuitry connected to the MCU pins, and upon subsystems such as the timer, serial peripheral interface (SPI), or serial communications interface (SCI) that were or were not active when the wait mode was entered.

SCP1 and SCP0 — SCI Baud Rate Prescaler Select Bits

These two bits select a prescale factor for the SCI baud rate generator that determines the highest possible baud rate.

Table 6-1. Baud Rate Prescale Selects

SCP1 and SCP0	Divide Internal Clock By	Crystal Frequency in MHz			
		4.0 MHz (Baud)	8.0 MHz (Baud)	10.0 MHz (Baud)	12.0 MHz (Baud)
0 0	1	62.50 K	125.0 K	156.25 K	187.5 K
0 1	3	20.83 K	41.67 K	52.08 K	62.5 K
1 0	4	15.625 K	31.25 K	38.4 K	46.88 K
1 1	13	4800	9600	12.02 K	14.42 K

SCR2–SCR0 — SCI Baud Rate Select Bits

These three bits select receiver and transmitter bit rate based on output from baud rate prescaler stage.

Table 6-2. Baud Rate Selects

SCR2–SCR0	Divide Prescaler By	Highest Baud Rate (Prescaler Output from Table 6-1)		
		4800	9600	38.4 K
0 0 0	1	4800	9600	38.4 K
0 0 1	2	2400	4800	19.2 K
0 1 0	4	1200	2400	9600
0 1 1	8	600	1200	4800
1 0 0	16	300	600	2400
1 0 1	32	150	300	1200
1 1 0	64	—	150	600
1 1 1	128	—	—	300

The prescale bits, SCP1 and SCP0, determine the highest baud rate and the SCR2–SCR0 bits select an additional binary submultiple (÷1, ÷2, ÷4, through ÷128) of this highest baud rate. The result of these two dividers in series is the 16 X receiver baud rate clock. The SCR2–SCR0 bits are not affected by reset and can be changed at any time, although they should not be changed when any SCI transfer is in progress.

Figure 6-8 illustrates the SCI baud rate timing chain. The prescale select bits determine the highest baud rate. The rate select bits determine additional divide by two stages to arrive at the receiver timing (RT) clock rate. The baud rate clock is the result of dividing the RT clock by 16.

Chapter 7
Serial Peripheral Interface (SPI)

7.1 Introduction

The serial peripheral interface (SPI), an independent serial communications subsystem, allows the microcontroller unit (MCU) to communicate synchronously with peripheral devices, such as:

- Transistor-transistor logic (TTL) shift registers
- Liquid crystal diode (LCD) display drivers
- Analog-to-digital converter (ADC) subsystems
- Other microprocessors (MCUs)

The SPI is also capable of inter-processor communication in a multiple master system. The SPI system can be configured as either a master or a slave device with data rates as high as one half of the E-clock rate when configured as master, and as fast as the E-clock rate when configured as slave.

7.2 Functional Description

The central element in the SPI system is the block containing the shift register and the read data buffer. The system is single buffered in the transmit direction and double buffered in the receive direction. This means that new data for transmission cannot be written to the shifter until the previous transfer is complete; however, received data is transferred into a parallel read data buffer so the shifter is free to accept a second serial character. As long as the first character is read out of the read data buffer before the next serial character is ready to be transferred, no overrun condition occurs. A single MCU register address is used for reading data from the read data buffer, and for writing data to the shifter.

The SPI status block represents the SPI status functions (transfer complete, write collision, and mode fault) performed by the serial peripheral status register (SPSR). The SPI control block represents those functions that control the SPI system through the serial peripheral control register (SPCR).

Refer to Figure 7-1, which shows the SPI block diagram.

7.3 SPI Transfer Formats

During an SPI transfer, data is simultaneously transmitted and received. A serial clock line synchronizes shifting and sampling of the information on the two serial data lines. A slave select line allows individual selection of a slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. On a master SPI device, the select line can optionally be used to indicate a multiple master bus contention. Refer to Figure 7-2.

8.3.1 Timer Control 2 Register

Use the control bits of timer control 2 register (TCTL2) to program input capture functions to detect a particular edge polarity on the corresponding timer input pin. Each of the input capture functions can be independently configured to detect rising edges only, falling edges only, any edge (rising or falling), or to disable the input capture function. The input capture functions operate independently of each other and can capture the same TCNT value if the input edges are detected within the same timer count cycle.

Address: $0021

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
Reset:	0	0	0	0	0	0	0	0

Figure 8-3. Timer Control 2 Register (TCTL2)

EDGxB and EDGxA — Input Capture Edge Control

There are four pairs of these bits. Each pair is cleared to 0 by reset and must be encoded to configure the corresponding input capture edge detector circuit. IC4 functions only if the I4/O5 bit in PACTL is set. Refer to Table 8-2 for timer control configuration.

Table 8-2. Timer Control Configuration

EDGxB	EDGxA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any edge

8.3.2 Timer Input Capture Registers

When an edge has been detected and synchronized, the 16-bit free-running counter value is transferred into the input capture register pair as a single 16-bit parallel transfer. Timer counter value captures and timer counter incrementing occur on opposite half-cycles of the phase two clock so that the count value is stable whenever a capture occurs. The timer input capture (TICx) registers are not affected by reset. Input capture values can be read from a pair of 8-bit read-only registers. A read of the high-order byte of an input capture register pair inhibits a new capture transfer for one bus cycle. If a double-byte read instruction, such as LDD, is used to read the captured value, coherency is assured. When a new input capture occurs immediately after a high-order byte read, transfer is delayed for an additional cycle but the value is not lost.

Address: $0010 — TIC1 (High)

	Bit 15	14	13	12	11	10	9	Bit 8
Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Write:								
Reset:	Unaffected by reset							

= Unimplemented

Figure 8-4. Timer Input Capture Registers (TICx)

DDRA7 — Data Direction Control for Port A Bit 7
Refer to 8.7 Pulse Accumulator.

PAEN — Pulse Accumulator System Enable Bit
Refer to 8.7 Pulse Accumulator.

PAMOD — Pulse Accumulator Mode Bit
Refer to 8.7 Pulse Accumulator.

PEDGE — Pulse Accumulator Edge Control Bit
Refer to 8.7 Pulse Accumulator.

DDRA3 — Data Direction Register for Port A Bit 3
Refer to Chapter 5 Input/Output (I/O) Ports.

I4/O5 — Input Capture 4/Output Compare 5 Bit
Refer to 8.3 Input Capture.

RTR1 and RTR0 — RTI Interrupt Rate Select Bits
These two bits determine the rate at which the RTI system requests interrupts. The RTI system is driven by an E divided by 2^{13} rate clock that is compensated so it is independent of the timer prescaler. These two control bits select an additional division factor. See Table 8-6.

Table 8-6. Real-Time Interrupt Rates

RTR1 and RTR0	E = 1 MHz	E = 2 MHz	E = 3 MHz	E = X MHz
0 0	2.731 ms	4.096 ms	8.192 ms	$(E/2^{13})$
0 1	5.461 ms	8.192 ms	16.384 ms	$(E/2^{14})$
1 0	10.923 ms	16.384 ms	32.768 ms	$(E/2^{15})$
1 1	21.845 ms	32.768 ms	65.536 ms	$(E/2^{16})$

8.6 Computer Operating Properly Watchdog Function

The clocking chain for the COP function, tapped off of the main timer divider chain, is only superficially related to the main timer system. The CR1 and CR0 bits in the OPTION register and the NOCOP bit in the CONFIG register determine the status of the COP function. Refer to Chapter 4 Resets, Interrupts, and Low-Power Modes for a more detailed discussion of the COP function.

8.7 Pulse Accumulator

The MC68HC711D3 has an 8-bit counter that can be configured to operate either as a simple event counter or for gated time accumulation, depending on the state of the PAMOD bit in the PACTL register. Refer to the pulse accumulator block diagram, Figure 8-19.

In the event counting mode, the 8-bit counter is clocked to increasing values by an external pin. The maximum clocking rate for the external event counting mode is the E clock divided by two. In gated time accumulation mode, a free-running E-clock ÷ 64 signal drives the 8-bit counter, but only while the external PAI pin is activated. Refer to Table 8-7. The pulse accumulator counter can be read or written at any time.

Pulse accumulator control bits are also located within two timer registers, TMSK2 and TFLG2, as described here.

8.7.1 Pulse Accumulator Control Register

Four of the pulse accumulator control register (PACTL) bits control an 8-bit pulse accumulator system. Another bit enables either the OC5 function or the IC4 function, while two other bits select the rate for the real-time interrupt system.

Address: $0026

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0
Reset:	0	0	0	0	0	0	0	0

Figure 8-20. Pulse Accumulator Control Register (PACTL)

DDRA7 — Data Direction Control for Port A Bit 7

The pulse accumulator uses port A bit 7 as the PAI input, but the pin can also be used as general-purpose I/O or as an output compare.

NOTE

Even when port A bit 7 is configured as an output, the pin still drives the input to the pulse accumulator.

Refer to Chapter 5 Input/Output (I/O) Ports for more information.

PAEN — Pulse Accumulator System Enable Bit

0 = Pulse accumulator disabled
1 = Pulse accumulator enabled

PAMOD — Pulse Accumulator Mode Bit

0 = Event counter
1 = Gated time accumulation

PEDGE — Pulse Accumulator Edge Control Bit

This bit has different meanings depending on the state of the PAMOD bit, as shown in Table 8-8.

Table 8-8. Pulse Accumulator Edge Control

PAMOD	PEDGE	Action on Clock
0	0	PAI falling edge increments the counter.
0	1	PAI rising edge increments the counter.
1	0	A 0 on PAI inhibits counting.
1	1	A 1 on PAI inhibits counting.

DDRA3 — Data Direction Register for Port A Bit 3

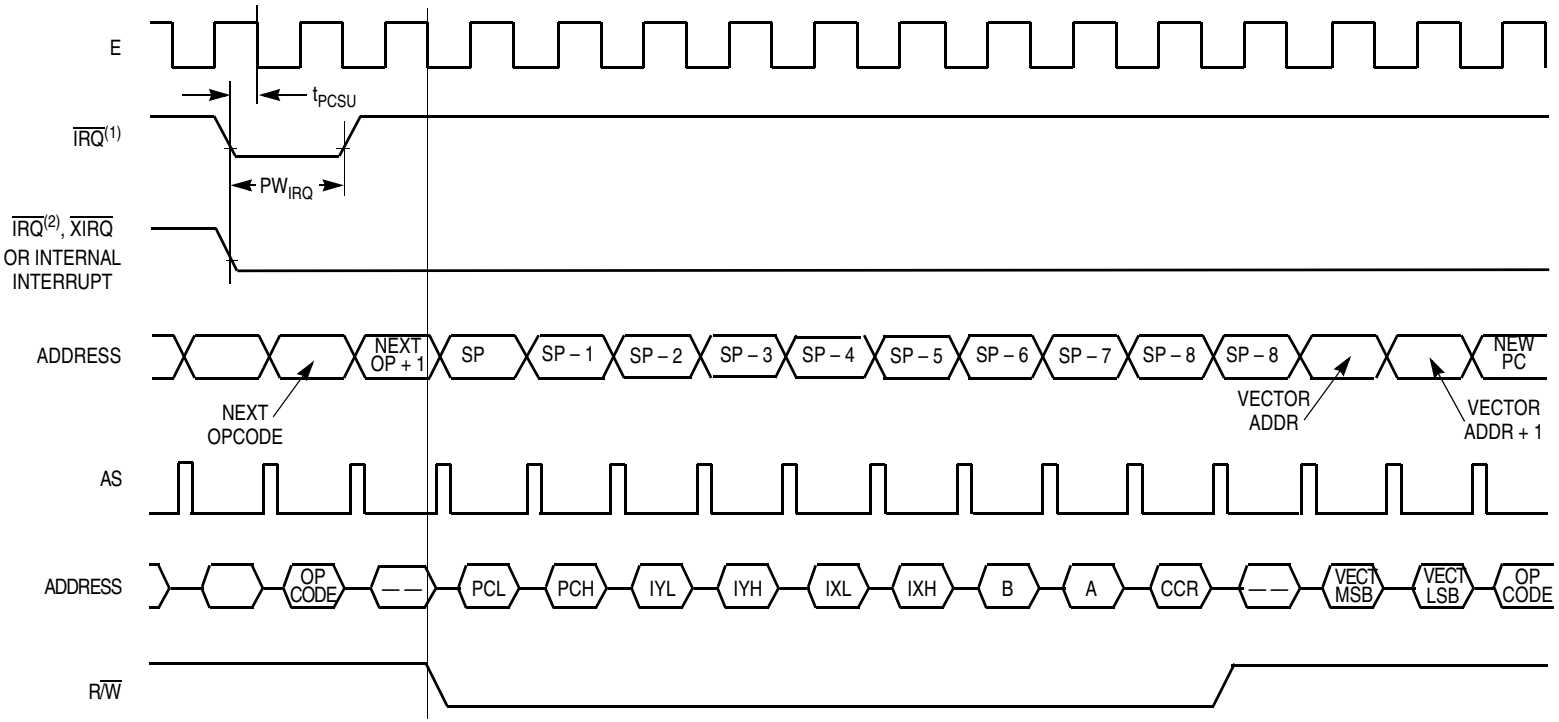
Refer to Chapter 5 Input/Output (I/O) Ports.

I4/O5 — Input Capture 4/Output Compare 5 Bit

Refer to 8.3 Input Capture.

RTR1 and RTR0 — RTI Interrupt Rate Select Bits

Refer to 8.5 Real-Time Interrupt.



Notes:
1. Edge sensitive $\overline{IRQ}$ pin (IRQE bit = 1)
2. Level sensitive $\overline{IRQ}$ pin (IRQE bit = 0)

Figure 9-7. Interrupt Timing Diagram

9.9 Serial Peripheral Interface Timing

Num	Characteristic[1]	Symbol	2.0 MHz		3.0 MHz		Unit
			Min	Max	Min	Max	
	Operating frequency Master Slave	 $f_{op(m)}$ $f_{op(s)}$	 dc dc	 0.5 2.0	 dc dc	 0.5 3.0	f_{op} MHz
1	Cycle time Master Slave	 $t_{cyc(m)}$ $t_{CYC(s)}$	 2.0 500	 — —	 2.0 333	 — —	t_{cyc} ns
2	Enable lead time Master[2] Slave	 $t_{lead(m)}$ $t_{lead(s)}$	 — 250	 — —	 — 240	 — —	ns
3	Enable lag time Master[2] Slave	 $t_{lag(m)}$ $t_{lag(s)}$	 — 250	 — —	 — 240	 — —	ns
4	Clock (SCK) high time Master Slave	 $t_{w(SCKH)m}$ $t_{w(SCKH)s}$	 340 190	 — —	 227 127	 — —	ns
5	Clock (SCK) low time Master Slave	 $t_{w(SCKL)m}$ $t_{w(SCKL)s}$	 340 190	 — —	 227 127	 — —	ns
6	Data setup time (inputs) Master Slave	 $t_{su(m)}$ $t_{su(s)}$	 100 100	 — —	 100 100	 — —	ns
7	Data hold time (inputs) Master Slave	 $t_{h(m)}$ $t_{h(s)}$	 100 100	 — —	 100 100	 — —	ns
8	Access time (time to data active from high-impedance state) Slave	t_a	0	120	0	120	ns
9	Disable time (hold time to high-impedance state) Slave	t_{dis}	—	240	—	167	ns
10	Data valid (after enable edge)[3]	$t_{v(s)}$	—	240	—	167	ns
11	Data hold time (outputs) (after enable edge)	t_{ho}	0	—	0	—	ns
12	Rise time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI outputs (SCK, MOSI, and MISO) SPI inputs (SCK, MOSI, MISO, and $\overline{SS}$)	 t_{rm} t_{rs}	 — —	 100 2.0	 — —	 100 2.0	 ns µs
13	Fall time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI outputs (SCK, MOSI, and MISO) SPI inputs (SCK, MOSI, MISO, and $\overline{SS}$)	 t_{fm} t_{fs}	 — —	 100 2.0	 — —	 100 2.0	 ns µs

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. Signal production depends on software.
3. Assumes 200 pF load on all SPI pins.

Chapter 10
Ordering Information and Mechanical Specifications

10.1 Introduction

This section provides ordering information for the MC68HC711D3. In addition, mechanical specifications are provided for the following packaging options:

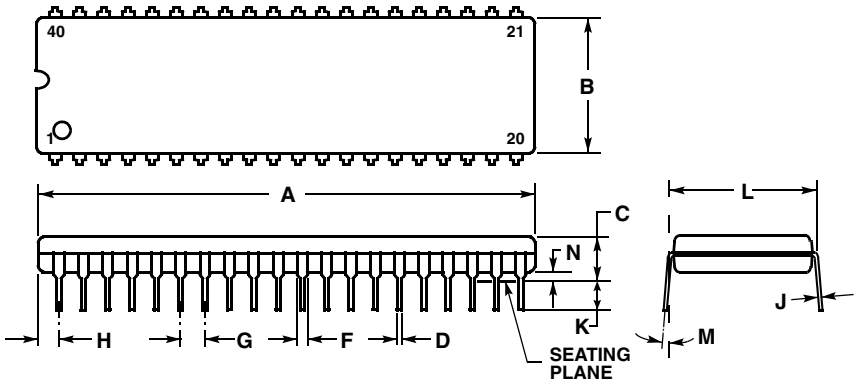
- 40-pin plastic dual in-line package (DIP)
- 44-pin plastic leaded chip carrier (PLCC)
- 44-pin plastic quad flat pack (QFP)

10.2 Ordering Information

Table 10-1. MC Order Numbers

Package Type	Temperature	MC Order Number	
		2 MHz	3 MHz
40-pin DIP	–40 to +85°C	MC68HC711D3CP2	MC68HC711D3CP3
44-pin PLCC	–40 to +85°C	MC68HC711D3CFN2	MC68HC711D3CFN3
	–40 to +105°C	MC68HC711D3VFN2	MC68HC711D3VFN3
44-pin QFP	–40 to +85°C	MC68HC711D3CFB2	MC68HC711D3CFB3

10.3 40-Pin DIP (Case 711-03)



NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	51.69	52.45	2.035	2.065
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

Appendix B
MC68L11D0

B.1 Introduction

The MC68L11D0 is an extended-voltage version of the MC68HC11D0 microcontroller that can operate in applications that require supply voltages as low as 3.0 volts. Operation is identical to that of the MC68HC11D0 (see Appendix A MC68HC11D3 and MC68HC11D0) in all aspects other than electrical parameters, as shown in this appendix.

Features of the MC68HC11D0 include:

- Suitable for battery-powered portable and hand-held applications
- Excellent for use in devices such as remote sensors and actuators
- Operating performance is same at 5 V and 3 V

B.2 MC68L11D0 Electrical Characteristics

The parameters given in Chapter 9 Electrical Characteristics apply to the MC68L11D0 with the exceptions given here.

B.2.1 Functional Operating Temperature Range

Rating	Symbol	Value	Unit
Operating temperature range	T_A	T_L to T_H –20 to +70	°C

B.2.2 DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Max	Unit
Output voltage[2] All outputs except XTAL I_{Load} = ± 10.0 µAAll outputs except XTAL, $\overline{RESET}$, and MODA	V_{OL} V_{OH}	— V_{DD} – 0.1	0.1 —	V
Output high voltage[1] All outputs except XTAL, $\overline{RESET}$, and MODA I_{Load} = – 0.5 mA, V_{DD} = 3.0 V I_{Load} = – 0.8 mA, V_{DD} = 4.5 V	V_{OH}	V_{DD} – 0.8	—	V
Output low voltage All outputs except XTAL I_{Load} = 1.6 mA, V_{DD} = 5.0 V I_{Load} = 1.0 mA, V_{DD} = 3.0 V	V_{OL}	—	0.4	V

The dc electrical table continues on next page.

B.2.6 Serial Peripheral Interface Timing

Num	Characteristic[1]	Symbol	1.0 MHz		2.0 MHz		Unit
			Min	Max	Min	Max	
	Operating frequency Master Slave	 $f_{op(m)}$ $f_{op(s)}$	 dc dc	 0.5 1.0	 dc dc	 0.5 2.0	f_{op} MHz
1	Cycle time Master Slave	 $t_{cyc(m)}$ $t_{CYC(s)}$	 2.0 1000	 — —	 2.0 500	 — —	t_{cyc} ns
2	Enable lead time Master[2] Slave	 $t_{lead(m)}$ $t_{lead(s)}$	 — 500	 — —	 — 250	 — —	ns
3	Enable lag time Master[2] Slave	 $t_{lag(m)}$ $t_{lag(s)}$	 — 500	 — —	 — 250	 — —	ns
4	Clock (SCK) high time Master Slave	 $t_{w(SCKH)m}$ $t_{w(SCKH)s}$	 680 380	 — —	 340 190	 — —	ns
5	Clock (SCK) low time Master Slave	 $t_{w(SCKL)m}$ $t_{w(SCKL)s}$	 680 380	 — —	 340 190	 — —	ns
6	Data setup time (inputs) Master Slave	 $t_{su(m)}$ $t_{su(s)}$	 100 100	 — —	 100 100	 — —	ns
7	Data hold time (inputs) Master Slave	 $t_{h(m)}$ $t_{h(s)}$	 100 100	 — —	 100 100	 — —	ns
8	Access time (time to data active from high-impedance state) Slave	t_a	0	120	0	120	ns
9	Disable time (hold time to high-impedance state) Slave	t_{dis}	—	240	—	240	ns
10	Data valid (after enable edge)[3]	$t_{v(s)}$	—	240	—	240	ns
11	Data hold time (outputs) (after enable edge)	t_{ho}	0	—	0	—	ns
12	Rise time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI outputs (SCK, MOSI, and MISO) SPI inputs (SCK, MOSI, MISO, and $\overline{SS}$)	 t_{rm} t_{rs}	 — —	 100 2.0	 — —	 100 2.0	 ns µs
13	Fall time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI outputs (SCK, MOSI, and MISO) SPI inputs (SCK, MOSI, MISO, and $\overline{SS}$)	 t_{fm} t_{fs}	 — —	 100 2.0	 — —	 100 2.0	 ns µs

1. V_{DD} = 3.0 Vdc to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. Signal production depends on software.
3. Assumes 100 pF load on all SPI pins.