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	30MHz
Connectivity	ATA, Compact Flash, EBI/EMI, Memory Card, SCI, SD, Smart Media, USB
Peripherals	POR, WDT
Number of I/O	41
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.5K x 8
Voltage - Supply (Vcc/Vdd)	2.25V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12uf32pb

List of Figures

List of Tables

- “Memory Stick Information for Developers,” ver. 1.3, 2000, Sony Corp.
- “The MultiMediaCard system specification,” ver. 3.0, 1/2001, MMCA Technical Committee.
- “SD Memory Card Specifications,” ver. 1.0, March 2000, SD Group.
- “Smart Media Electrical Specifications,” ver. 1.0, May 19, 1999, SSFDC Forum Technical Committee.
- “Universal Serial Bus Specification,” rev. 2.0, 27 April 2000, Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips.

Part Number

Figure 0-1 provides an ordering number example.

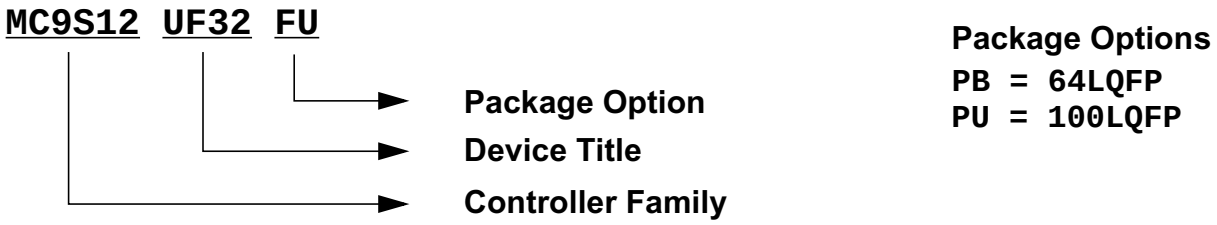


Figure 0-1 Order Part Number Coding

Table 0-2 lists the part number coding based on the package.

Table 0-2 Part Number Coding

Part Number	Package	Description
MC9S12UF32PB	64LQFP	MC9S12UF32
MC9S12UF32PU	100LQFP	MC9S12UF32

Section 1 Introduction

1.1 Overview

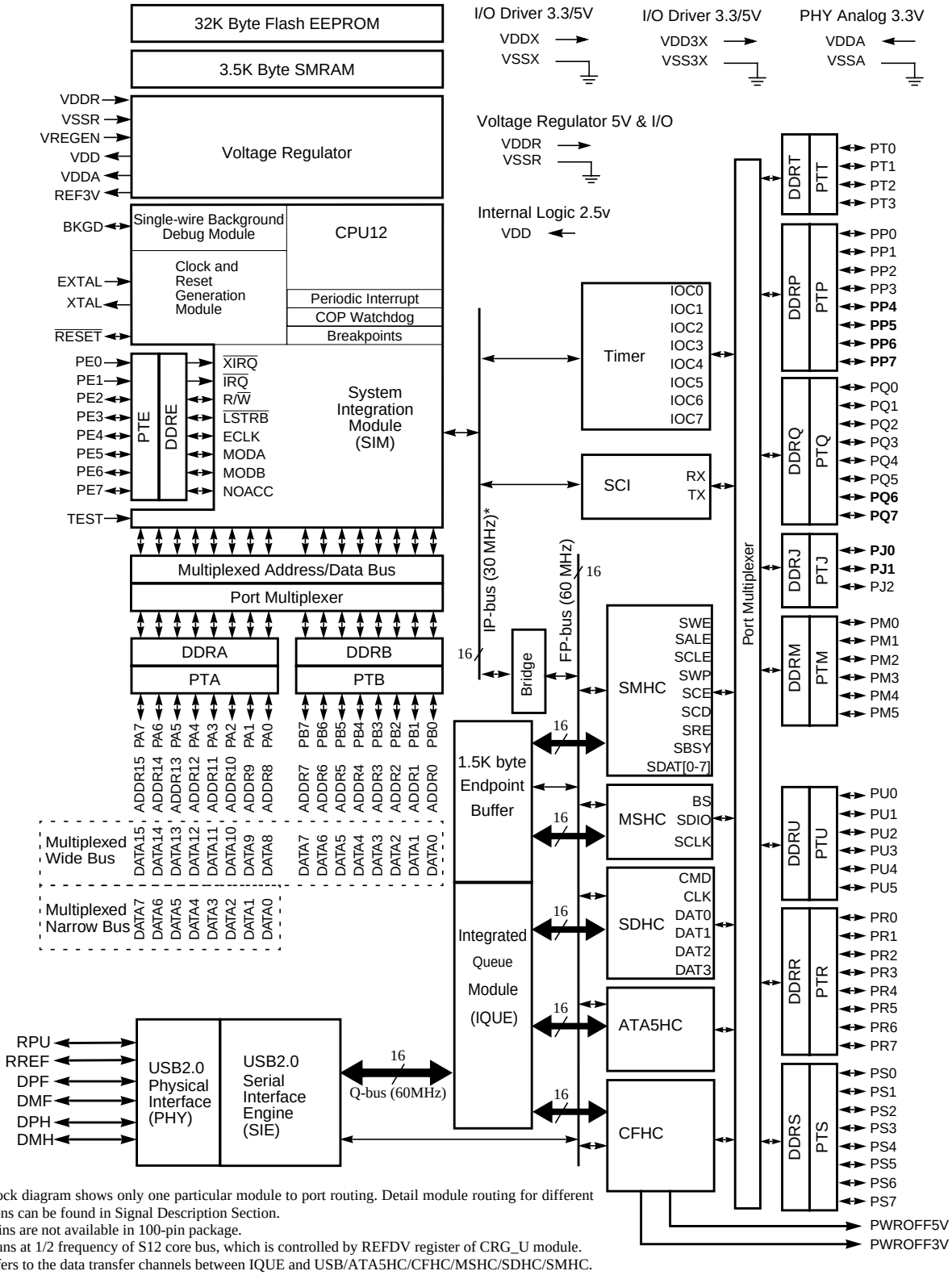
The MC9S12UF32 microcontroller unit (MCU) is USB2.0 device for memory card reader and ATA/ATAPI interface applications. This device is composed of standard on-chip modules including a 16-bit central processing unit (HCS12 CPU), 32k bytes of Flash EEPROM, 3.5k bytes of RAM, USB2.0 interface, Integrated Queue Controller (IQUE) block with 1.5k bytes RAM buffer for USB Bulk data transport, ATA5 interface, Compact Flash interface, SD/MMC interface, SmartMedia interface, Memorystick interface, a 16-bit 8-channel timer, Serial Communication Interface, 73 discrete digital I/O channels and 2 input only channels[1]. The MC9S12UF32 has full 16-bit internal data paths throughout.

1.2 Features

- HCS12 Core
  - 16-bit HCS12 CPU
    i. Upward compatible with M68HC11 instruction set
    ii. Interrupt stacking and programmer's model identical to M68HC11
    iii. Instruction queue
    iv. Enhanced indexed addressing
  - Multiplexed External Bus Interface (MEBI)
  - Memory Mapping Control (MMC)
  - Interrupt Control (INT)
  - Single-wire Background Debug Mode (BDM)
  - On-chip hardware Breakpoints (BKP)
- Clock and Reset Generator (CRG_U)
  - Clock Throttle to prescale the oscillator clock or 60Mhz clock from USB20D6E2F.
  - COP watchdog
  - Real Time Interrupt
- Memory
  - 32K Flash EEPROM
    - Internal program/erase voltage generation
    - Security and Block Protect bits
  - 3.5K byte RAM

NOTES:
1. Not all functions are available simultaneously.

1.4 Block Diagram



* This block diagram shows only one particular module to port routing. Detail module routing for different applications can be found in Signal Description Section.
* Some pins are not available in 100-pin package.
* IPbus runs at 1/2 frequency of S12 core bus, which is controlled by REFDV register of CRG_U module.
*Qbus refers to the data transfer channels between IQUE and USB/ATA5HC/CFHC/MSHC/SDHC/SMHC.

Figure 1-1 MC9S12UF32 Block Diagram

$00C8 - $00CF SCI (Asynchronous Serial Interface)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$00CD	SCISR2	Read:	0	0	0	0	0	BRK13	TXDIR	RAF
		Write:								
$00CE	SCIDRH	Read:	R8	T8	0	0	0	0	0	0
		Write:								
$00CF	SCIDRL	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0

$00D0 - $00FF Reserved

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$00D0 - $00FF	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								

$0100 - $010F Flash Control Register

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$0100	FCLKDIV	Read:	FDIVLD	PRDIV8	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
		Write:								
$0101	FSEC	Read:	KEYEN1	KEYEN0	NV5	NV4	NV3	NV2	SEC1	SEC0
		Write:								
$0102	FTSTMOD	Read:	0	0	0	WRALL	0	0	0	0
		Write:								
$0103	FCNFG	Read:	CBEIE	CCIE	KEYACC	0	0	0	BKSEL1	BKSEL0
		Write:								
$0104	FPROT	Read:	FPOPEN	NV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0
		Write:								
$0105	FSTAT	Read:	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	0	0
		Write:								
$0106	FCMD	Read:	0	CMDB6	CMDB5	0	0	CMDB2	0	CMDB0
		Write:								
$0107	Reserved for Factory Test	Read:	0	0	0	0	0	0	0	0
		Write:								
$0108	Reserved for Factory Test	Read:	0	0	0	0	0	0	0	0
		Write:								
$0109	Reserved for Factory Test	Read:	0	0	0	0	0	0	0	0
		Write:								
$010A	Reserved for Factory Test	Read:	0	0	0	0	0	0	0	0
		Write:								
$010B	Reserved for Factory Test	Read:	0	0	0	0	0	0	0	0
		Write:								
$010C - $010F	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								

$0110 - $011B Reserved

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$0110 - $011B	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								

$011C - $011F SMRAM Control Register

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$011C	SMRAMCFG	Read:	0	0	0	0	0	0	0	SMMODE
		Write:								
$011D	SMRAMSTAT	Read:	0	0	0	0	0	0	0	PSMBA
		Write:								
$011E - $011F	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								

$0120 - $01BF Reserved

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$0120 - $01BF	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								

$01C0 - $01FF ATA Host Controller (ATA5HC)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$01C0	HCFG (hi)	Read:	SMR	FR	0	0	CLK_EN	XNW	IE	IORDY_EN
		Write:								
$01C1	HCFG (lo)	Read:	0	0	0	0	0	0	0	0
		Write:								
$01C2	HSR (hi)	Read:	TIP	UREP	DRAB	0	FF	FE	RERR	WERR
		Write:								
$01C3	HSR (lo)	Read:	0	0	0	0	0	0	0	0
		Write:								
$01C4	HPIO1 (hi)	Read:	0	PIO_T0						
		Write:								
$01C5	HPIO1 (lo)	Read:	0	0	PIO_T2_8					
		Write:								
$01C6	HPIO2 (hi)	Read:	0	0	PIO_T2_16					
		Write:								
$01C7	HPIO2 (lo)	Read:	0	0	0	0	PIO_T2I			
		Write:								
$01C8	HPIO3 (hi)	Read:	0	0	0	0	0	PIO_T4		
		Write:								
$01C9	HPIO3 (lo)	Read:	0	0	0	0	PIO_T1			
		Write:								
$01CA	HPIO4 (hi)	Read:	0	0	0	0	0	PIO_TA		
		Write:								
$01CB	HPIO4 (lo)	Read:	0	0	0	0	0	0	0	0
		Write:								

$01C0 - $01FF ATA Host Controller (ATA5HC)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$01E5	HUDMA9 (lo)	Read:	0	0	0	0	0	0	0	0
		Write:								
$01E6 - $01ED	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								
$01EE	DCTR/DASR[1] (hi)	Read:	BSY	DRDY	#		DRQ	obs		ERR
		Write:						SRST	nIEN	ZERO
$01EF	DCTR/DASR[1] (lo)	Read:	0	0	0	0	0	0	0	0
		Write:								
$01F0	DDR (hi)[1]	Read:	BYTE_E							
		Write:								
$01F1	DDR (lo)[1]	Read:	BYTE_O							
		Write:								
$01F2	DFR/DER (hi)[1]	Read:	#r1					ABRT	#r2	
		Write:	#w							
$01F3	DFR/DER (lo)[1]	Read:	0	0	0	0	0	0	0	0
		Write:								
$01F4	DSCR (hi)[1]	Read:	#							
		Write:								
$01F5	DSCR (lo)[1]	Read:	0	0	0	0	0	0	0	0
		Write:								
$01F6	DSNR (hi)[1]	Read:	#							
		Write:								
$01F7	DSNR (lo)[1]	Read:	0	0	0	0	0	0	0	0
		Write:								
$01F8	DCLR (hi)[1]	Read:	#							
		Write:								
$01F9	DCLR (lo)[1]	Read:	0	0	0	0	0	0	0	0
		Write:								
$01FA	DCHR (hi)[1]	Read:	#							
		Write:								
$01FB	DCHR (lo)[1]	Read:	0	0	0	0	0	0	0	0
		Write:								
$01FC	DDHR (hi)[1]	Read:	obs	#1	obs	DEV	#2			
		Write:								
$01FD	DDHR (lo)[1]	Read:	0	0	0	0	0	0	0	0
		Write:								
$01FE	DCR/DSR (hi)[1]	Read:	BSY	DRDY	#r		DRQ	obs		ERR
		Write:	#w							
$01FF	HDMAM (lo)	Read:	PIE	HUT	AF	0	IE	UDMA	RD	WR
		Write:								

NOTES:

1. These registers are mapped to the registers on an external ATA/ATAPI device, for detail explanation of the #, #r1, #r2, #w, #1,#2 field, please refer to the ATAHC block guide and ATA/ATAPI standards.

2.4.8 DPF - USB Full Speed D+ data line

DPF is the D+ analog input output line for full speed data communication in the USB physical layer module. This line is also used for D+ termination during high speed operation. Refer to USB20D6E2F block guide for further information.

2.4.9 DPH - USB High Speed D+ data line

DPH is the D+ analog input output line for high speed data communication in the USB physical layer module. This line will be high impedance during full speed operation. Refer to USB20D6E2F block guide for further information.

2.4.10 DMF - USB Full Speed D- data line

DMF is the D- analog input output line for full speed data communication in the USB physical layer module. This line is also used for D+ termination during high speed operation. Refer to USB20D6E2F block guide for further information.

2.4.11 DMH - USB High Speed D- data line

DMH is the D- analog input output line for high speed data communication in the USB physical layer module. This line will be high impedance during full speed operation. Refer to USB20D6E2F block guide for further information.

2.4.12 PWROFF5V - power off 5V supply

PWROFF5V is used for turning off 5V supply to external chips or memory card in conjunction with an external PMOS device, this signal is controlled by CFPMR register in CFHC module. Refer to CFHC block guide for further information.

2.4.13 PWROFF3V - power off 3V supply

PWROFF3V is used for turning off 3V supply to external chips or memory card in conjunction with an external PMOS device, this signal is controlled by CFPMR register in CFHC module. Refer to CFHC block guide for further information.

2.4.14 REF3V - 3.3V reference for external regulator

REF3V a regulator reference for driving an external NMOS device to provide the system with a regulated 3.3V supply. The feedback path for the REF3V is the VDD3X supply pin. Refer to VREG_U block guide for further information.

2.4.15 PA[7:0] / ADDR[15:8] / DATA[15:8] / CFD[15:8] / ATAD[15:8] — Port A

2.4.21 PE3 / $\overline{\text{LSTRB}}$ / $\overline{\text{TAGLO}}$ — Port E I/O Pin 3 / Low-Byte Strobe ($\overline{\text{LSTRB}}$)

PE3 can be used as a general-purpose I/O in all modes and is an input with an active pull-up out of reset. PE3 can also be configured as a Low-Byte Strobe ($\overline{\text{LSTRB}}$). The $\overline{\text{LSTRB}}$ signal is used in write operations, so external low byte writes will not be possible until this function is enabled. $\overline{\text{LSTRB}}$ can be enabled by setting the LSTRE bit in the PEAR register. In Expanded Wide and Emulation Narrow modes, and when BDM tagging is enabled, the $\overline{\text{LSTRB}}$ function is multiplexed with the $\overline{\text{TAGLO}}$ function. When enabled a logic zero on the $\overline{\text{TAGLO}}$ pin at the falling edge of ECLK will tag the low byte of an instruction word being read into the instruction queue.

2.4.22 PE2 / R/$\overline{\text{W}}$ — Port E I/O Pin 2 / Read/$\overline{\text{Write}}$

PE2 can be used as a general-purpose I/O in all modes and is configured an input with an active pull-up out of reset. If the read/$\overline{\text{write}}$ function is required it should be enabled by setting the RDWE bit in the PEAR register. External writes will not be possible until the read/$\overline{\text{write}}$ function is enabled.

2.4.23 PE1 / $\overline{\text{IRQ}}$ — Port E input Pin 1 / Maskable Interrupt Pin

PE1 is always an input and can always be read. The PE1 pin is also the $\overline{\text{IRQ}}$ input used for requesting an asynchronous interrupt to the MCU. During reset, the I bit in the condition code register (CCR) is set and any $\overline{\text{IRQ}}$ interrupt is masked until software enables it by clearing the I bit. The $\overline{\text{IRQ}}$ is software programmable to either falling edge-sensitive triggering or level-sensitive triggering based on the setting of the IRQE bit in the IRQCR register. The $\overline{\text{IRQ}}$ is always enabled and configured to level-sensitive triggering out of reset. It can be disabled by clearing IRQEN bit in the IRQCR register. There is an active pull-up on this pin while in reset and immediately out of reset. The pull-up can be turned off by clearing PUPEE in the PUCR register.

2.4.24 PE0 / $\overline{\text{XIRQ}}$ — Port E input Pin 0 / Non Maskable Interrupt Pin

PE0 is always an input and can always be read. The PE0 pin is also the $\overline{\text{XIRQ}}$ input for requesting a non-maskable asynchronous interrupt to the MCU. During reset, the X bit in the condition code register (CCR) is set and any $\overline{\text{XIRQ}}$ interrupt is masked until MCU software enables it by clearing the X bit. Because the $\overline{\text{XIRQ}}$ input is level sensitive triggered, it can be connected to a multiple-source wired-OR network. There is an active pull-up on this pin while in reset and immediately out of reset. The pull-up can be turned off by clearing PUPEE in the PUCR register.

2.4.25 PJ2 / MSSCLK/ROMCTL - Port J I/O Pin 2

PJ2 is a general purpose input or output pin. In expanded modes the PJ2 pin can be used to determine the reset state of the ROMON bit in the MISC register. At the rising edge of RESET, the state of the PJ2 pin is latched to the ROMON bit. When the MSHC module is enabled it becomes the serial clock line (MSSCLK) for the MSHC module. While in reset and immediately out of reset the PJ2 pin is configured as a high impedance input pin. Consult the Port Integration Module (PIM) PIM_9UF32 Block Guide and the MSHC Block Guide for information about pin configurations.

on-chip serial ports. After the backdoor sequence has been correctly matched, the microcontroller will be unsecured, and all Flash commands will be enabled and the Flash security byte can be programmed to the unsecure state, if desired.

Please note that if the system goes through a reset condition prior to successful configuration of unsecured mode the system will reset back into secured mode operation.

4.4.3.2 Unsecuring the Microcontroller (full FLASH erase)

In order to unsecure the microcontroller, the internal FLASH must be erased. This can be done through an external program in expanded mode.

Once the user has erased the FLASH, the part can be reset into special single chip mode This invokes a program that verifies the erasure of the internal FLASH. Once this program completes, the user can erase and program the FLASH security bits to the unsecured state. This is generally done through the BDM, but the user could also change to expanded mode (by writing the mode bits through the BDM) and jumping to an external program (again through BDM commands). Note that if the part goes through a reset before the security bits are reprogrammed to the unsecure state, the part will be secured again.

4.5 Low Power Modes

There are two low power modes available on the MC9S12UF32: Stop and Wait

Please see **Table A-8** for device operating characteristics in Stop and Wait modes. Consult the CRG_U Block Guide and the respective Block Guide for information on the module behavior in Stop and Wait Mode.

4.5.1 Stop

Executing the CPU STOP instruction stops all clocks and the oscillator, thus putting the chip in fully static mode. Wake up from this mode can be done via reset or external interrupts.

4.5.2 Wait

This mode is entered by executing the CPU WAI instruction. In this mode, the CPU will not execute instructions. The internal CPU signals (address and data bus) will be fully static. All peripherals stay active. So, for example, data can still be transferred from ATA5HC to USB via IQUE. For further power consumption, the peripherals can individually turn off their local clocks.

4.5.3 Run

Although this is not a low power mode, unused peripheral modules should not be enabled in order to save power.

$$P_{INT} = I_{DDR} \cdot V_{DDR}$$

I_{DDR} is the current shown in **Table A-8** and not the overall current flowing into VDDR, which additionally contains the current flowing into the external loads with output high.

$$P_{IO} = \sum_{i} R_{DSON} \cdot I_{IO_i}^{2}$$

Which is the sum of all output currents on I/O ports associated with VDD3X, VDDX and VDDR.

Table A-5 Thermal Package Characteristics[1]

Num	C	Rating	Symbol	Min	Typ	Max	Unit
1	T	Thermal Resistance LQFP100, single layer PCB natural convection	θ_{JA}	-	56		$^{o}C/W$
2	T	Thermal Resistance LQFP100, four layer PCB natural convection	θ_{JA}	-	46		$^{o}C/W$
3	T	Thermal Resistance LQFP64, single layer PCB natural convection.	θ_{JA}	-	64		$^{o}C/W$
4	T	Thermal Resistance LQFP64, four layer PCB natural convection.	θ_{JA}	-	48		$^{o}C/W$

NOTES:
1. The values for thermal resistance are achieved by package simulations

A.2 NVM, Flash

NOTE: *Unless otherwise noted the abbreviation NVM (Non Volatile Memory) is used for Flash.*

A.2.1 NVM timing

The time base for all NVM program or erase operations is derived from the oscillator. A minimum oscillator frequency f_{NVMOSC} is required for performing program or erase operations. The NVM modules do not have any means to monitor the frequency and will not prevent program or erase operation at frequencies above or below the specified minimum. Attempting to program or erase the NVM modules at a lower frequency a full program or erase transition is not assured.

The Flash program and erase operations are timed using a clock derived from the oscillator using the FCLKDIV register. The frequency of this clock must be set within the limits specified as f_{NVMOP}.

The minimum program and erase times shown in **Table A-9** are calculated for maximum f_{NVMOP} and maximum f_{bus}. The maximum times are calculated for minimum f_{NVMOP} and a f_{bus} of 2MHz.

A.2.1.1 Single Word Programming

The programming time for single word programming is dependant on the bus frequency as a well as on the frequency f_{NVMOP} and can be calculated according to the following formula.

$$t_{swpgm} = 9 \cdot \frac{1}{f_{NVMOP}} + 25 \cdot \frac{1}{f_{bus}}$$

A.2.1.2 Burst Programming

This applies only to the Flash where up to 32 words in a row can be programmed consecutively using burst programming by keeping the command pipeline filled. The time to program a consecutive word can be calculated as:

$$t_{bwpgm} = 4 \cdot \frac{1}{f_{NVMOP}} + 9 \cdot \frac{1}{f_{bus}}$$

The time to program a whole row is:

$$t_{brpgm} = t_{swpgm} + 31 \cdot t_{bwpgm}$$

Burst programming is more than 2 times faster than single word programming.

A.2.1.3 Sector Erase

Erasing a 512 byte Flash sector takes:

$$t_{era} \approx 4000 \cdot \frac{1}{f_{NVMOP}}$$

The setup time can be ignored for this operation.

A.2.1.4 Mass Erase

Erasing a NVM block takes:

$$t_{mass} \approx 20000 \cdot \frac{1}{f_{NVMOP}}$$

The setup time can be ignored for this operation.

A.2.1.5 Blank Check

The time it takes to perform a blank check on the Flash is dependant on the location of the first non-blank word starting at relative address zero. It takes one bus cycle per word to verify plus a setup of the command.

$$t_{check} \approx location \cdot t_{cyc} + 10 \cdot t_{cyc}$$

Table A-9 NVM Timing Characteristics

Conditions are shown in **Table A-4** unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	D	External Oscillator Clock	f_{NVMOSC}	0.5		60 [1]	MHz
2	D	Bus frequency for Programming or Erase Operations	f_{NVMBUS}	1			MHz
3	D	Operating Frequency	f_{NVMOP}	150		200	kHz
4	P	Single Word Programming Time	t_{swpgm}	46 [2]		74.5 [3]	µs
5	D	Flash Burst Programming consecutive word	t_{bwpgm}	20.4 [2]		31 [3]	µs
6	D	Flash Burst Programming Time for 32 Words	t_{brpgm}	678.4 [2]		1035.5 [3]	µs
7	P	Sector Erase Time	t_{era}	20 [4]		26.7 [3]	ms
8	P	Mass Erase Time	t_{mass}	100 [4]		133 [3]	ms
9	D	Blank Check Time Flash per block	t_{check}	11 [5]		32778 [6]	t_{cyc}

NOTES:
1. Restrictions for oscillator in crystal mode apply!
2. Minimum Programming times are achieved under maximum NVM operating frequency f_{NVMOP} and maximum bus frequency f_{bus}.
3. Maximum Erase and Programming times are achieved under particular combinations of f_{NVMOP} and bus frequency f_{bus}. Refer to formulae in Sections **Section A.2.1.1 Single Word Programming- Section A.2.1.4 Mass Erase** for guidance.
4. Minimum Erase times are achieved under maximum NVM operating frequency f_{NVMOP}.
5. Minimum time, if first word in the array is not blank
6. Maximum time to complete check on an erased block

A.4.2 Oscillator

The device features an internal Pierce oscillator. The $\overline{XCLKS}$ signal of the OSC module is tied internally that the Pierce oscillator/external clock mode is always selected. Pierce oscillator/external clock mode allows the input of a square wave.

.

Table A-13 Oscillator Characteristics

Conditions are shown in **Table A-4** unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	C	Crystal oscillator range (Pierce) [1]	f_{OSC}	0.5	12	40	MHz
2	P	Startup Current	i_{OSC}	100			µA
3	P	External square wave input frequency	f_{EXT}	0.5		60	MHz
4	D	External square wave pulse width low	t_{EXTL}	8			ns
5	D	External square wave pulse width high	t_{EXTH}	8			ns
6	D	External square wave rise time	t_{EXTR}			1	ns
7	D	External square wave fall time	t_{EXTF}			1	ns
8	D	Input Capacitance (EXTAL, XTAL pins)	C_{IN}		7		pF
9	P	EXTAL Pin Input High Voltage	$V_{IH,EXTAL}$	$0.75*V_{DD}$			V
	T	EXTAL Pin Input High Voltage	$V_{IH,EXTAL}$			V_{DD} + 0.3	V
10	P	EXTAL Pin Input Low Voltage	$V_{IL,EXTAL}$			$0.25*V_{DD}$	V
	T	EXTAL Pin Input Low Voltage	$V_{IL,EXTAL}$	V_{SSA} - 0.3			V
11	C	EXTAL Pin Input Hysteresis	$V_{HYS,EXTAL}$		250		mV

NOTES:

1. Depending on the crystal a damping series resistor might be necessary. Also, the oscillator frequency must be 12MHz in order to have the USB interface and the storage interface to work properly.

A.4.3 USB PHY

The oscillator provides the reference clock for the USB PHY. The 12MHz oscillator clock is used for the USB PHY to generate the 480bps USB traffic, 30MHz USB UTMI interface clock and the 60MHz internal clock. An internal circuitry checks the generated 60MHz clock against the input 12MHz oscillator clock. The generated 60MHz clock is assumed correct (locked) if there are exactly 15 clock cycles in 3 oscillator clock cycles.

For the electrical characteristics of the USB PHY, please refer to Chapter 7 “Electrical” of Universal Serial Bus Specification Revision 2.0.

Table A-15 Expanded Bus Timing Characteristics

Conditions are shown in **Table A-4** unless otherwise noted, C_{LOAD} = 50pF							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	P	Frequency of operation (E-clock)	f_o	0		30.0	MHz
2	P	Cycle time	t_{cyc}	33			ns
3	D	Pulse width, E low	PW_{EL}	16			ns
4	D	Pulse width, E high[1]	PW_{EH}	16			ns
5	D	Address delay time	t_{AD}			5	ns
6	D	Address valid time to E rise (PW_{EL}–t_{AD})	t_{AV}	11			ns
7	D	Muxed address hold time	t_{MAH}	2			ns
8	D	Address hold to data valid	t_{AHDS}	7			ns
9	D	Data hold to address	t_{DHA}	2			ns
10	D	Read data setup time	t_{DSR}	13			ns
11	D	Read data hold time	t_{DHR}	0			ns
12	D	Write data delay time	t_{DDW}			7	ns
13	D	Write data hold time	t_{DHW}	2			ns
14	D	Write data setup time(1) (PW_{EH}–t_{DDW})	t_{DSW}	12			ns
15	D	Address access time(1) (t_{cyc}–t_{AD}–t_{DSR})	t_{ACCA}	15			ns
16	D	E high access time(1) (PW_{EH}–t_{DSR})	t_{ACCE}	3			ns
17	D	Read/write delay time	t_{RWD}			4	ns
18	D	Read/write valid time to E rise (PW_{EL}–t_{RWD})	t_{RWV}	12			ns
29	D	Read/write hold time	t_{RWH}	2			ns
20	D	Low strobe delay time	t_{LSD}			4	ns
21	D	Low strobe valid time to E rise (PW_{EL}–t_{LSD})	t_{LSV}	12			ns
22	D	Low strobe hold time	t_{LSH}	2			ns
23	D	NOACC strobe delay time	t_{NOD}			4	ns
24	D	NOACC valid time to E rise (PW_{EL}–t_{NOD})	t_{NOV}	12			ns
25	D	NOACC hold time	t_{NOH}	2			ns
26	D	IPIPO[1:0] delay time	t_{P0D}	2		4	ns
27	D	IPIPO[1:0] valid time to E rise (PW_{EL}–t_{P0D})	t_{P0V}	11			ns
28	D	IPIPO[1:0] delay time(1) (PW_{EH}-t_{P1V})	t_{P1D}	2		25	ns
29	D	IPIPO[1:0] valid time to E fall	t_{P1V}	11			ns

NOTES:
1. Affected by clock stretch: add N x t_{cyc} where N=0,1,2 or 3, depending on the number of clock stretches.

System on a Chip Guide End Sheet

FINAL PAGE OF
128
PAGES