NXP USA Inc. - S912XHZ256F1VAG Datasheet

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	Not For New Designs
Core Processor	HCS12X
Core Size	16-Bit
Speed	80MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, LINbus, SCI, SPI
Peripherals	LCD, Motor control PWM, POR, PWM, WDT
Number of I/O	117
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 16x8/10b
Oscillator Type	External
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912xhz256f1vag

Table of Contents

Chapter 1
MC9S12XHZ Family Device Overview

1.1 Introduction 21
- 1.1.1 Features 22
- 1.1.2 Modes of Operation 24
- 1.1.3 Block Diagram 24
- 1.1.4 Device Memory Map 26
- 1.1.5 Part ID Assignments 30

1.2 Signal Description 30
- 1.2.1 Device Pinout 30
- 1.2.2 Signal Properties Summary 33
- 1.2.3 Detailed Signal Descriptions 36
- 1.2.4 Power Supply Pins 45

1.3 System Clock Description 47

1.4 Chip Configuration Summary 48

1.5 Modes of Operation 49
- 1.5.1 User Modes 49
- 1.5.2 Low-Power Modes 50
- 1.5.3 Freeze Mode 51

1.6 Resets and Interrupts 51
- 1.6.1 Vectors 51
- 1.6.2 Effects of Reset 54

1.7 COP Configuration 54

1.8 ATD External Trigger Input Connection 55

Chapter 2
Port Integration Module (S12XHZPIMV1)

2.1 lntroduction 57
- 2.1.1 Features 57
- 2.1.2 Block Diagram 58

2.2 External Signal Description 59

2.3 Memory Map and Register Definition 65
- 2.3.1 Port A and Port B 68
- 2.3.2 Port C and Port D 71
- 2.3.3 Port E 73
- 2.3.4 Port K 75
- 2.3.5 Miscellaneous registers 77
- 2.3.6 Port AD 82

Chapter 23
External Bus Interface (S12XEBIV3)

23.1 Introduction 807
23.1.1 Glossary or Terms 808
23.1.2 Features 808
23.1.3 Modes of Operation 808
23.1.4 Block Diagram 810
23.2 External Signal Description 810
23.3 Memory Map and Register Definition 812
23.3.1 Module Memory Map 812
23.3.2 Register Descriptions 812
23.4 Functional Description 816
23.4.1 Operating Modes and External Bus Properties 816
23.4.2 Internal Visibility 817
23.4.3 Accesses to Port Replacement Registers 821
23.4.4 Stretched External Bus Accesses 821
23.4.5 Data Select and Data Direction Signals 822
23.4.6 Low-Power Options 824
23.5 Initialization/Application Information 824
23.5.1 Normal Expanded Mode 825
23.5.2 Emulation Modes 826

Chapter 24
Interrupt (S12XINTV1)

24.1 Introduction 831
24.1.1 Glossary 832
24.1.2 Features 832
24.1.3 Modes of Operation 832
24.1.4 Block Diagram 834
24.2 External Signal Description 835
24.3 Memory Map and Register Definition 835
24.3.1 Register Descriptions 836
24.4 Functional Description 843
24.4.1 S12X Exception Requests 843
24.4.2 Interrupt Prioritization 843
24.4.3 XGATE Requests 844
24.4.4 Priority Decoders 844
24.4.5 Reset Exception Requests 845
24.4.6 Exception Priority 845
24.5 Initialization/Application Information 846
24.5.1 Initialization 846
24.5.2 Interrupt Nesting 846

Table 2-1. Detailed Signal Descriptions (Sheet 4 of 6)

Port	Pin Name	Pin Function and Priority	I/O	Description	Pin Function after Reset
S	PS[7]	$\overline{\text{SS}}$	I/O	Serial peripheral interface slave select input/output in master mode, input in slave mode	GPIO
		GPIO	I/O	General-purpose I/O	
	PS[6]	SCK	I/O	Serial peripheral interface serial clock pin	
		GPIO	I/O	General-purpose I/O	
	PS[5]	MOSI	I/O	Serial peripheral interface master out/slave in pin	
		GPIO	I/O	General-purpose I/O	
	PS[4]	MISO	I/O	Serial peripheral interface master in/slave out pin	
		GPIO	I/O	General-purpose I/O	
	PS[3]	TXD1	O	Serial communication interface 1 transmit pin	
		GPIO	I/O	General-purpose I/O	
	PS[2]	$\overline{\text{CS3}}$	O	Chip select 3	
		RXD1	I	Serial communication interface 1 receive pin	
		GPIO	I/O	General-purpose I/O	
	PS[1]	TXD0	O	Serial communication interface 0 transmit pin	
		GPIO	I/O	General-purpose I/O	
	PS[0]	RXD0	I	Serial communication interface 0 receive pin	
		GPIO	I/O	General-purpose I/O	
T	PT[7]	IOC7	I/O	Timer channel	GPIO
		SCL1	I/O	Inter-integrated circuit 1 serial clock line	
		GPIO	I/O	General-purpose I/O	
	PT[6]	IOC7	I/O	Timer channel	
		SDA1	I/O	Inter-integrated circuit 1 serial data line	
		GPIO	I/O	General-purpose I/O	
	PT[5]	IOC5	I/O	Timer channel	
		SCL0	I/O	Inter-integrated circuit 0 serial clock line	
		GPIO	I/O	General-purpose I/O	
	PT[4]	IOC4	I/O	Timer channel	
		SDA0	I/O	Inter-integrated circuit 0 serial data line	
		GPIO	I/O	General-purpose I/O	
	PT[3:0]	FP[27:24]	I/O	LCD frontplane driver	
		IOC[3:0]	I/O	Timer channel	
		GPIO	I/O	General-purpose I/O	

3.3.2.5.1 Flash Protection Restrictions

The general guideline is that Flash protection can only be added and not removed. Table 3-14 specifies all valid transitions between Flash protection scenarios. Any attempt to write an invalid scenario to the FPROT register will be ignored and the FPROT register will remain unchanged. The contents of the FPROT register reflect the active protection scenario. See the FPHS and FPLS descriptions for additional restrictions.

Table 3-14. Flash Protection Scenario Transitions

From Protection Scenario	To Protection Scenario[1]							
	0	1	2	3	4	5	6	7
0	X	X	X	X				
1		X		X				
2			X	X				
3				X				
4				X	X			
5			X	X	X	X		
6		X		X	X		X	
7	X	X	X	X	X	X	X	X

[1] Allowed transitions marked with X.

3.3.2.6 Flash Status Register (FSTAT)

The FSTAT register defines the operational status of the module.

Module Base + 0x0005

	7	6	5	4	3	2	1	0
R	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	0	0
W		▨			▨	▨	▨	▨
Reset	1	1	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 3-12. Flash Status Register (FSTAT — Normal Mode)

Module Base + 0x0005

	7	6	5	4	3	2	1	0
R	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	FAIL	0
W		▨			▨	▨		▨
Reset	1	1	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 3-13. Flash Status Register (FSTAT — Special Mode)

4.3.2.7 EEPROM Command Register (ECMD)

The ECMD register is the EEPROM command register.

Module Base + 0x0006

	7	6	5	4	3	2	1	0
R	0	CMDB						
W	▨							
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 4-11. EEPROM Command Register (ECMD)

All CMDB bits are readable and writable during a command write sequence while bit 7 reads 0 and is not writable.

Table 4-7. ECMD Field Descriptions

Field	Description
6:0 CMDB[6:0]	**EEPROM Command Bits** — Valid EEPROM commands are shown in Table 4-8. Writing any command other than those listed in Table 4-8 sets the ACCERR flag in the ESTAT register.

Table 4-8. Valid EEPROM Command List

CMDB[6:0]	Command
0x05	Erase Verify
0x20	Word Program
0x40	Sector Erase
0x41	Mass Erase
0x47	Sector Erase Abort
0x60	Sector Modify

4.3.2.8 RESERVED3

This register is reserved for factory testing and is not accessible.

Module Base + 0x0007

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 4-12. RESERVED3

		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0018 XGSWTM	R	0	0	0	0	0	0	0	0	XGSWT[7:0]							
	W	XGSWTM[7:0]															
0x001A XGSEMM	R	0	0	0	0	0	0	0	0	XGSEM[7:0]							
	W	XGSEMM[7:0]															
0x001C Reserved	R									▨	▨	▨	▨	▨	▨	▨	▨
	W																
0x001D XGCCR	R									0	0	0	0	XGN	XGZ	XGV	XGC
	W									▨	▨	▨	▨				
0x001E XGPC	R	XGPC															
	W																
0x0020 Reserved	R									▨	▨	▨	▨	▨	▨	▨	▨
	W																
0x0021 Reserved	R									▨	▨	▨	▨	▨	▨	▨	▨
	W																
0x0022 XGR1	R	XGR1															
	W																
0x0024 XGR2	R	XGR2															
	W																
0x0026 XGR3	R	XGR3															
	W																
0x0028 XGR4	R	XGR4															
	W																
0x002A XGR5	R	XGR5															
	W																
0x002C XGR6	R	XGR6															
	W																
0x002E XGR7	R	XGR7															
	W																

▨ = Unimplemented or Reserved

Figure 5-2. XGATE Register Summary (Sheet 3 of 3)

CMP Compare CMP

Operation

RS1 – RS2 ⇒ NONE (translates to SUB R0, RS1, RS2)
RD – IMM16 ⇒ NONE (translates to CMPL RD, #IMM16[7:0]; CPCH RD, #IMM16[15:8])

Subtracts two 16 bit values and discards the result.

CCR Effects

N	Z	V	C
Δ	Δ	Δ	Δ

N: Set if bit 15 of the result is set; cleared otherwise.

Z: Set if the result is $0000; cleared otherwise.

V: Set if a two´s complement overflow resulted from the operation; cleared otherwise.
$\mathrm{RS1[15]} \,\&\, \overline{\mathrm{RS2[15]}} \,\&\, \overline{\mathrm{result[15]}} \mid \overline{\mathrm{RS1[15]}} \,\&\, \mathrm{RS2[15]} \,\&\, \mathrm{result[15]}$
$\mathrm{RD[15]} \,\&\, \overline{\mathrm{IMM16[15]}} \,\&\, \overline{\mathrm{result[15]}} \mid \overline{\mathrm{RD[15]}} \,\&\, \mathrm{IMM16[15]} \,\&\, \mathrm{result[15]}$

C: Set if there is a carry from the bit 15 of the result; cleared otherwise.
$\overline{\mathrm{RS1[15]}} \,\&\, \mathrm{RS2[15]} \mid \overline{\mathrm{RS1[15]}} \,\&\, \mathrm{result[15]} \mid \mathrm{RS2[15]} \,\&\, \mathrm{result[15]}$
$\overline{\mathrm{RD[15]}} \,\&\, \mathrm{IMM16[15]} \mid \overline{\mathrm{RD[15]}} \,\&\, \mathrm{result[15]} \mid \mathrm{IMM16[15]} \,\&\, \mathrm{result[15]}$

Code and CPU Cycles

Source Form	Address Mode	Machine Code										Cycles
CMP RS1, RS2	TRI	0	0	0	1	1	0 0 0	RS1	RS2	0	0	P
CMP RS, #IMM16	IMM8	1	1	0	1	0	RS	IMM16[7:0]				P
	IMM8	1	1	0	1	1	RS	IMM16[15:8]				P

COM One’s Complement COM

Operation

~RS $\Rightarrow$ RD (translates to XNOR RD, R0, RS)
~RD $\Rightarrow$ RD (translates to XNOR RD, R0, RD)

Performs a one’s complement on a general purpose register.

CCR Effects

N	Z	V	C
Δ	Δ	0	—

N: Set if bit 15 of the result is set; cleared otherwise.
Z: Set if the result is $0000; cleared otherwise.
V: 0; cleared.
C: Not affected.

Code and CPU Cycles

Source Form	Address Mode	Machine Code										Cycles
COM RD, RS	TRI	0	0	0	1	0	RD	0 0 0	RS	1	1	P
COM RD	TRI	0	0	0	1	0	RD	0 0 0	RD	1	1	P

Special single chip erase and unsecure sequence:

1. Reset into special single chip mode.
2. Write an appropriate value to the ECLKDIV register for correct timing.
3. Write 0xFF to the EPROT register to disable protection.
4. Write 0x30 to the ESTAT register to clear the PVIOL and ACCERR bits.
5. Write 0x0000 to the EDATA register (0x011A–0x011B).
6. Write 0x0000 to the EADDR register (0x0118–0x0119).
7. Write 0x41 (mass erase) to the ECMD register.
8. Write 0x80 to the ESTAT register to clear CBEIF.
9. Write an appropriate value to the FCLKDIV register for correct timing.
10. Write 0x00 to the FCNFG register to select Flash block 0.
11. Write 0x10 to the FTSTMOD register (0x0102) to set the WRALL bit, so the following writes affect all Flash blocks.
12. Write 0xFF to the FPROT register to disable protection.
13. Write 0x30 to the FSTAT register to clear the PVIOL and ACCERR bits.
14. Write 0x0000 to the FDATA register (0x010A–0x010B).
15. Write 0x0000 to the FADDR register (0x0108–0x0109).
16. Write 0x41 (mass erase) to the FCMD register.
17. Write 0x80 to the FSTAT register to clear CBEIF.
18. Wait until all CCIF flags are set.
19. Reset back into special single chip mode.
20. Write an appropriate value to the FCLKDIV register for correct timing.
21. Write 0x00 to the FCNFG register to select Flash block 0.
22. Write 0xFF to the FPROT register to disable protection.
23. Write 0xFFBE to Flash address 0xFF0E.
24. Write 0x20 (program) to the FCMD register.
25. Write 0x80 to the FSTAT register to clear CBEIF.
26. Wait until the CCIF flag in FSTAT is are set.
27. Reset into any mode.

7.3.2.6 CRG Clock Select Register (CLKSEL)

This register controls CRG clock selection. Refer to Figure 7-17 for more details on the effect of each bit.

Module Base +0x_05

	7	6	5	4	3	2	1	0
R	PLLSEL	PSTP	0	0	PLLWAI	0	RTIWAI	COPWAI
W			▨	▨		▨		
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 7-9. CRG Clock Select Register (CLKSEL)

Read: Anytime

Write: Refer to each bit for individual write conditions

Table 7-4. CLKSEL Field Descriptions

Field	Description
7 PLLSEL	**PLL Select Bit** — Write anytime. Writing a1 when LOCK = 0 and AUTO = 1, or TRACK = 0 and AUTO = 0 has no effect This prevents the selection of an unstable PLLCLK as SYSCLK. PLLSEL bit is cleared when the MCU enters self clock mode, Stop mode or wait mode with PLLWAI bit set. **It is recommended to read back the PLLSEL bit to make sure PLLCLK has really been selected as SYSCLK, as LOCK status bit could theoretically change at the very moment writing the PLLSEL bit.** 0 System clocks are derived from OSCCLK ($f_{BUS} = f_{OSC} / 2$). 1 System clocks are derived from PLLCLK ($f_{BUS} = f_{PLL} / 2$).
6 PSTP	**Pseudo Stop Bit** Write: Anytime This bit controls the functionality of the oscillator during stop mode. 0 Oscillator is disabled in stop mode. 1 Oscillator continues to run in stop mode (pseudo stop). **Note:** Pseudo stop mode allows for faster STOP recovery and reduces the mechanical stress and aging of the resonator in case of frequent STOP conditions at the expense of a slightly increased power consumption.
3 PLLWAI	**PLL Stops in Wait Mode Bit** Write: Anytime If PLLWAI is set, the CRG will clear the PLLSEL bit before entering wait mode. The PLLON bit remains set during wait mode, but the PLL is powered down. Upon exiting wait mode, the PLLSEL bit has to be set manually if PLL clock is required. While the PLLWAI bit is set, the AUTO bit is set to 1 in order to allow the PLL to automatically lock on the selected target frequency after exiting wait mode. 0 PLL keeps running in wait mode. 1 PLL stops in wait mode.
1 RTIWAI	**RTI Stops in Wait Mode Bit** Write: Anytime 0 RTI keeps running in wait mode. 1 RTI stops and initializes the RTI dividers whenever the part goes into wait mode.
0 COPWAI	**COP Stops in Wait Mode Bit** Normal modes: Write once Special modes: Write anytime 0 COP keeps running in wait mode. 1 COP stops and initializes the COP counter whenever the part goes into wait mode.

Table 11-2. MC10B12C - Memory Map

$23	MCDC1 (low byte)	RW
$24	MCDC2 (high byte)	RW
$25	MCDC2 (low byte)	RW
$26	MCDC3 (high byte)	RW
$27	MCDC3 (low byte)	RW
$28	MCDC4 (high byte)	RW
$29	MCDC4 (low byte)	RW
$2A	MCDC5 (high byte)	RW
$2B	MCDC5 (low byte)	RW
$2C	MCDC6 (high byte)	RW
$2D	MCDC6 (low byte)	RW
$2E	MCDC7 (high byte)	RW
$2F	MCDC7 (low byte)	RW
$30	MCDC8 (high byte)	RW
$31	MCDC8 (low byte)	RW
$32	MCDC9 (high byte)	RW
$33	MCDC9 (low byte)	RW
$34	MCDC10 (high byte)	RW
$35	MCDC10 (low byte)	RW
$36	MCDC11 (high byte)	RW
$37	MCDC11 (low byte)	RW
$38	Reserved	-
$39	Reserved	-
$3A	Reserved	-
$3B	Reserved	-
$3C	Reserved	-
$3D	Reserved	-
$3E	Reserved	-
$3F	Reserved	-

Figure 14-40 shows how the first 32-bit filter bank (CANIDAR0–CANIDAR3, CANIDMR0–CANIDMR3) produces a filter 0 hit. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces a filter 1 hit.

- Four identifier acceptance filters, each to be applied to:
  - The 14 most significant bits of the extended identifier plus the SRR and IDE bits of CAN 2.0B messages.
  - The 11 bits of the standard identifier, the RTR and IDE bits of CAN 2.0A/B messages.
    Figure 14-41 shows how the first 32-bit filter bank (CANIDAR0–CANIDAR3, CANIDMR0–CANIDMR3) produces filter 0 and 1 hits. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces filter 2 and 3 hits.
- Eight identifier acceptance filters, each to be applied to the first 8 bits of the identifier. This mode implements eight independent filters for the first 8 bits of a CAN 2.0A/B compliant standard identifier or a CAN 2.0B compliant extended identifier.
  Figure 14-42 shows how the first 32-bit filter bank (CANIDAR0–CANIDAR3, CANIDMR0–CANIDMR3) produces filter 0 to 3 hits. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces filter 4 to 7 hits.
- Closed filter. No CAN message is copied into the foreground buffer RxFG, and the RXF flag is never set.

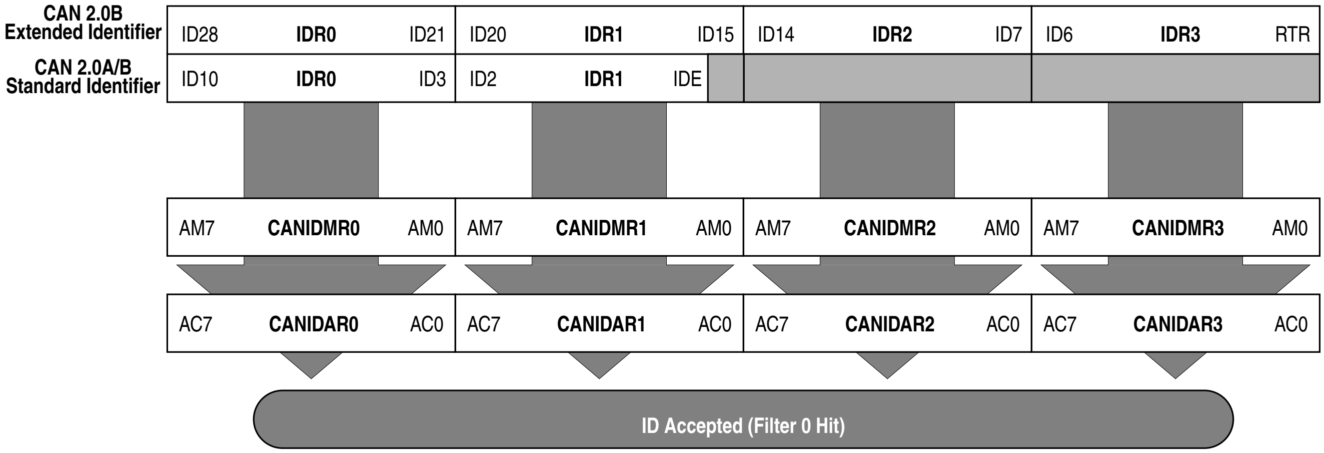

Figure 14-40. 32-bit Maskable Identifier Acceptance Filter

Each channel counter can be read at anytime without affecting the count or the operation of the PWM channel.

Any value written to the counter causes the counter to reset to $00, the counter direction to be set to up, the immediate load of both duty and period registers with values from the buffers, and the output to change according to the polarity bit. When the channel is disabled (PWMEx = 0), the counter stops. When a channel becomes enabled (PWMEx = 1), the associated PWM counter continues from the count in the PWMCNTx register. This allows the waveform to continue where it left off when the channel is re-enabled. When the channel is disabled, writing "0" to the period register will cause the counter to reset on the next selected clock.

NOTE

If the user wants to start a new "clean" PWM waveform without any "history" from the old waveform, the user must write to channel counter (PWMCNTx) prior to enabling the PWM channel (PWMEx = 1).

Generally, writes to the counter are done prior to enabling a channel in order to start from a known state. However, writing a counter can also be done while the PWM channel is enabled (counting). The effect is similar to writing the counter when the channel is disabled, except that the new period is started immediately with the output set according to the polarity bit.

NOTE

Writing to the counter while the channel is enabled can cause an irregular PWM cycle to occur.

The counter is cleared at the end of the effective period (see Section 18.4.2.5, "Left Aligned Outputs" and Section 18.4.2.6, "Center Aligned Outputs" for more details).

Table 18-10. PWM Timer Counter Conditions

Counter Clears ($00)	Counter Counts	Counter Stops
When PWMCNTx register written to any value	When PWM channel is enabled (PWMEx = 1). Counts from last value in PWMCNTx.	When PWM channel is disabled (PWMEx = 0)
Effective period ends		

18.4.2.5 Left Aligned Outputs

The PWM timer provides the choice of two types of outputs, left aligned or center aligned. They are selected with the CAEx bits in the PWMCAE register. If the CAEx bit is cleared (CAEx = 0), the corresponding PWM output will be left aligned.

In left aligned output mode, the 8-bit counter is configured as an up counter only. It compares to two registers, a duty register and a period register as shown in the block diagram in Figure 18-19. When the PWM counter matches the duty register the output flip-flop changes state causing the PWM waveform to also change state. A match between the PWM counter and the period register resets the counter and the output flip-flop, as shown in Figure 18-19, as well as performing a load from the double buffer period and duty register to the associated registers, as described in Section 18.4.2.3, "PWM Period and Duty". The counter counts from 0 to the value in the period register – 1.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x002B ICSYS	R	SH37	SH26	SH15	SH04	TFMOD	PACMX	BUFEN	LATQ
	W								
0x002C OCPD	R	OCPD7	OCPD6	OCPD5	OCPD4	OCPD3	OCPD2	OCPD1	OCPD0
	W								
0x002D TIMTST	R	Timer Test Register							
	W								
0x002E PTPSR	R	PTPS7	PTPS6	PTPS5	PTPS4	PTPS3	PTPS2	PTPS1	PTPS0
	W								
0x002F PTMCPSR	R	PTMPS7	PTMPS6	PTMPS5	PTMPS4	PTMPS3	PTMPS2	PTMPS1	PTMPS0
	W								
0x0030 PBCTL	R	0	PBEN	0	0	0	0	PBOVI	0
	W	▨		▨	▨	▨	▨		▨
0x0031 PBFLG	R	0	0	0	0	0	0	PBOVF	0
	W	▨	▨	▨	▨	▨	▨		▨
0x0032 PA3H	R	PA3H7	PA3H6	PA3H5	PA3H4	PA3H3	PA3H2	PA3H1	PA3H0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0033 PA2H	R	PA2H7	PA2H6	PA2H5	PA2H4	PA2H3	PA2H2	PA2H1	PA2H0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0034 PA1H	R	PA1H7	PA1H6	PA1H5	PA1H4	PA1H3	PA1H2	PA1H1	PA1H0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0035 PA0H	R	PA0H7	PA0H6	PA0H5	PA0H4	PA0H3	PA0H2	PA0H1	PA0H0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0036 MCCNT (High)	R	MCCNT15	MCCNT14	MCCNT13	MCCNT12	MCCNT11	MCCNT10	MCCNT9	MCCNT8
	W								
0x0037 MCCNT (Low)	R	MCCNT7	MCCNT6	MCCNT5	MCCNT4	MCCNT3	MCCNT2	MCCNT1	MCCNT0
	W								
0x0038 TC0H (High)	R	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0039 TC0H (Low)	R	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
		▨	▨	▨	▨	▨	▨	▨	▨

▨ = Unimplemented or Reserved

Figure 19-2. ECT Register Summary (Sheet 4 of 5)

19.3.2.16 Pulse Accumulator A Flag Register (PAFLG)

Module Base + 0x0021

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	PAOVF	PAIF
W	▨	▨	▨	▨	▨	▨		
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 19-37. Pulse Accumulator A Flag Register (PAFLG)

Read: Anytime

Write used in the flag clearing mechanism. Writing a one to the flag clears the flag. Writing a zero will not affect the current status of the bit.

NOTE

> When TFFCA = 1, the flags cannot be cleared via the normal flag clearing mechanism (writing a one to the flag). Reference Section 19.3.2.6, “Timer System Control Register 1 (TSCR1)”.

All bits reset to zero.

PAFLG indicates when interrupt conditions have occurred. The flags can be cleared via the normal flag clearing mechanism (writing a one to the flag) or via the fast flag clearing mechanism (Reference TFFCA bit in Section 19.3.2.6, “Timer System Control Register 1 (TSCR1)”).

Table 19-22. PAFLG Field Descriptions

Field	Description
1 PAOVF	**Pulse Accumulator A Overflow Flag** — Set when the 16-bit pulse accumulator A overflows from 0xFFFF to 0x0000, or when 8-bit pulse accumulator 3 (PAC3) overflows from 0x00FF to 0x0000. When PACMX = 1, PAOVF bit can also be set if 8-bit pulse accumulator 3 (PAC3) reaches 0x00FF followed by an active edge on IC3.
0 PAIF	**Pulse Accumulator Input edge Flag** — Set when the selected edge is detected at the IC7 input pin. In event mode the event edge triggers PAIF and in gated time accumulation mode the trailing edge of the gate signal at the IC7 input pin triggers PAIF.

19.3.2.17 Pulse Accumulators Count Registers (PACN3 and PACN2)

Module Base + 0x0022

	7	6	5	4	3	2	1	0
R W	PACNT7(15)	PACNT6(14)	PACNT5(13)	PACNT4(12)	PACNT3(11)	PACNT2(10)	PACNT1(9)	PACNT0(8)
Reset	0	0	0	0	0	0	0	0

Figure 19-38. Pulse Accumulators Count Register 3 (PACN3)

Table 19-23. MCCTL Field Descriptions (continued)

Field	Description
4 ICLAT	**Input Capture Force Latch Action** — When input capture latch mode is enabled (LATQ and BUFEN bit in ICSYS are set), a write one to this bit immediately forces the contents of the input capture registers TC0 to TC3 and their corresponding 8-bit pulse accumulators to be latched into the associated holding registers. The pulse accumulators will be automatically cleared when the latch action occurs. Writing zero to this bit has no effect. Read of this bit will always return zero.
3 FLMC	**Force Load Register into the Modulus Counter Count Register** — This bit is active only when the modulus down-counter is enabled (MCEN = 1). A write one into this bit loads the load register into the modulus counter count register (MCCNT). This also resets the modulus counter prescaler. Write zero to this bit has no effect. Read of this bit will return always zero.
2 MCEN	**Modulus Down-Counter Enable** 0 Modulus counter disabled. The modulus counter (MCCNT) is preset to 0xFFFF. This will prevent an early interrupt flag when the modulus down-counter is enabled. 1 Modulus counter is enabled.
1:0 MCPR[1:0]	**Modulus Counter Prescaler Select** — These two bits specify the division rate of the modulus counter prescaler when PRNT of TSCR1 is set to 0. The newly selected prescaler division rate will not be effective until a load of the load register into the modulus counter count register occurs.

Table 19-24. Modulus Counter Prescaler Select

MCPR1	MCPR0	Prescaler Division
0	0	1
0	1	4
1	0	8
1	1	16

19.3.2.20 16-Bit Modulus Down-Counter FLAG Register (MCFLG)

Module Base + 0x0027

	7	6	5	4	3	2	1	0
R	MCZF	0	0	0	POLF3	POLF2	POLF1	POLF0
W		▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 19-43. 16-Bit Modulus Down-Counter FLAG Register (MCFLG)

Read: Anytime

Write only used in the flag clearing mechanism for bit 7. Writing a one to bit 7 clears the flag. Writing a zero will not affect the current status of the bit.

Table 20-10. Interrupt Vectors

Interrupt Source	Local Enable
Autonomous periodical interrupt (API)	APIE = 1

20.4.10.1 Low-Voltage Interrupt (LVI)

In FPM, VREG_3V3 monitors the input voltage V_{DDA}. Whenever V_{DDA} drops below level $V_{LVIA,}$ the status bit LVDS is set to 1. On the other hand, LVDS is reset to 0 when V_{DDA} rises above level V_{LVID}. An interrupt, indicated by flag LVIF = 1, is triggered by any change of the status bit LVDS if interrupt enable bit LVIE = 1.

NOTE

On entering the reduced power mode, the LVIF is not cleared by the VREG_3V3.

20.4.10.2 Autonomous Periodical Interrupt (API)

As soon as the configured timeout period of the API has elapsed, the APIF bit is set. An interrupt, indicated by flag APIF = 1, is triggered if interrupt enable bit APIE = 1.

2 data entries, thus in this case the DBGCNT[0] is incremented after each separate entry. In Detail mode DBGCNT[0] remains cleared whilst the other DBGCNT bits are incremented on each trace buffer entry.

XGATE and CPU12X COFs occur independently of each other and the profile of COFs for the two sources is totally different. When both sources are being traced in Normal or Loop1 mode, for each COF from one source, there may be many COFs from the other source, depending on user code. COF events could occur far from each other in the time domain, on consecutive cycles or simultaneously. When a COF occurs in either source (S12X or XGATE) a trace buffer entry is made and the corresponding CDV or XDV bit is set. The current PC of the other source is simultaneously stored to the trace buffer even if no COF has occurred, in which case CDV/XDV remains cleared indicating the address is not associated with a COF, but is simply a snapshot of the PC contents at the time of the COF from the other source.

Single byte data accesses in Detail Mode are always stored to the low byte of the trace buffer (CDATAL or XDATAL) and the high byte is cleared. When tracing word accesses, the byte at the lower address is always stored to trace buffer byte3 and the byte at the higher address is stored to byte2

Table 22-43. Trace Buffer Organization

Mode	8-Byte Wide Word Buffer							
	7	6	5	4	3	2	1	0
XGATE Detail	CXINF1	CADRH1	CADRM1	CADRL1	XDATAH1	XDATAL1	XADRM1	XADRL1
	CXINF2	CADRH2	CADRM2	CADRL2	XDATAH2	XDATAL2	XADRM2	XADRL2
CPU12X Detail	CXINF1	CADRH1	CADRM1	CADRL1	CDATAH1	CDATAL1	XADRM1	XADRL1
	CXINF2	CADRH2	CADRM2	CADRL2	CDATAH2	CDATAL2	XADRM2	XADRL2
Both Other Modes	XINF0		XPCM0	XPCL0	CINF0	CPCH0	CPCM0	CPCL0
	XINF1		XPCM1	XPCL1	CINF1	CPCH1	CPCM1	CPCL1
XGATE Other Modes	XINF1		XPCM1	XPCL1	XINF0		XPCM0	XPCL0
	XINF3		XPCM3	XPCL3	XINF2		XPCM2	XPCL2
CPU12X Other Modes	CINF1	CPCH1	CPCM1	CPCL1	CINF0	CPCH0	CPCM0	CPCL0
	CINF3	CPCH3	CPCM3	CPCL3	CINF2	CPCH2	CPCM2	CPCL2

Table 23-17. Access in Normal Expanded Mode

Access	$\overline{\text{RE}}$	$\overline{\text{WE}}$	$\overline{\text{UDS}}$	$\overline{\text{LDS}}$	DATA[15:8]		DATA[7:0]	
					I/O	**data(addr)**	**I/O**	**data(addr)**
Word write of data on DATA[15:0] at an even and even+1 address	1	0	0	0	Out	data(even)	Out	data(odd)
Byte write of data on DATA[7:0] at an odd address	1	0	1	0	In	x	Out	data(odd)
Byte write of data on DATA[15:8] at an even address	1	0	0	1	Out	data(even)	In	x
Word read of data on DATA[15:0] at an even and even+1 address	0	1	0	0	In	data(even)	In	data(odd)
Byte read of data on DATA[7:0] at an odd address	0	1	1	0	In	x	In	data(odd)
Byte read of data on DATA[15:8] at an even address	0	1	0	1	In	data(even)	In	x
Indicates No Access	1	1	1	1	In	x	In	x
Unimplemented	1	1	1	0	In	x	In	x
	1	1	0	1	In	x	In	x

23.4.5.2 Emulation Modes and Special Test Mode

In emulation modes and special test mode, the external signals $\overline{\text{LSTRB}}$, R$\overline{\text{W}}$, and ADDR0 indicate the access type (read/write), data size and alignment of an external bus access. Misaligned accesses to the internal RAM and misaligned XGATE PRR accesses in emulation modes are the only type of access that are able to produce $\overline{\text{LSTRB}}$ = ADDR0 = 1. This is summarized in Table 23-18.

Table 23-18. Access in Emulation Modes and Special Test Mode

Access	R$\overline{\text{W}}$	$\overline{\text{LSTRB}}$	ADDR0	DATA[15:8]		DATA[7:0]	
				I/O	**data(addr)**	**I/O**	**data(addr)**
Word write of data on DATA[15:0] at an even and even+1 address	0	0	0	Out	data(even)	Out	data(odd)
Byte write of data on DATA[7:0] at an odd address	0	0	1	In	x	Out	data(odd)
Byte write of data on DATA[15:8] at an even address	0	1	0	Out	data(odd)	In	x
Word write at an odd and odd+1 internal RAM address (misaligned — only in emulation modes)	0	1	1	Out	data(odd+1)	Out	data(odd)
Word read of data on DATA[15:0] at an even and even+1 address	1	0	0	In	data(even)	In	data(even+1)
Byte read of data on DATA[7:0] at an odd address	1	0	1	In	x	In	data(odd)
Byte read of data on DATA[15:8] at an even address	1	1	0	In	data(even)	In	x
Word read at an odd and odd+1 internal RAM address (misaligned - only in emulation modes)	1	1	1	In	data(odd+1)	In	data(odd)

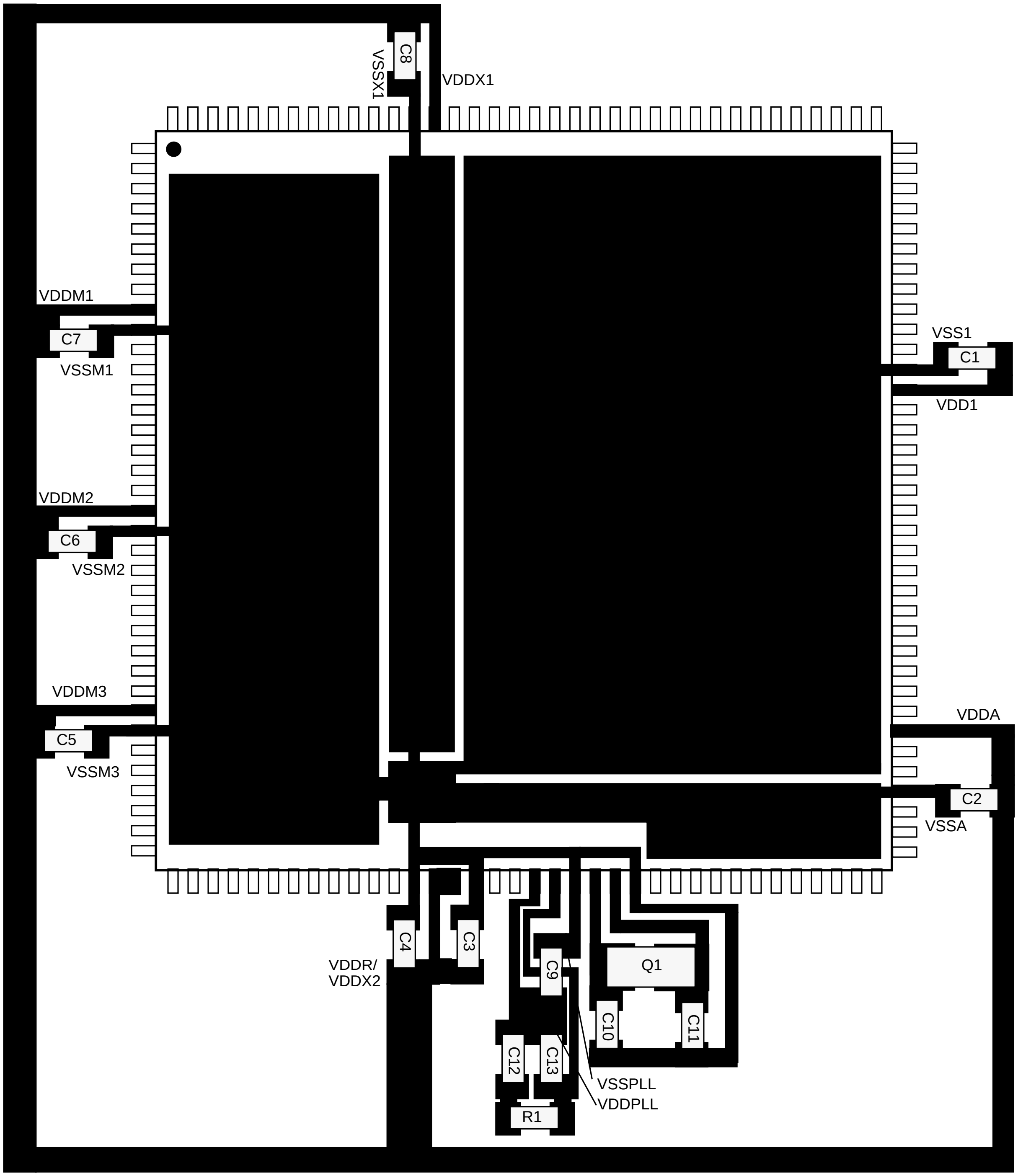

Figure C-1. LQFP144 Recommended PCB Layout