MC9S12P128 FCS
Full Chip Simulation Implementation Note
Covers MC9S12P Family

HCS12

CodeWarrior Tools

MC9S12P128 FCS

Rev. 1.00

2/2008

freescale.com

freescale semiconductor

1 Introduction

1.1 Overview of MC9S12P Family

The MC9S12P-Family of micro controllers units (MCU) are 16-bit devices composed of standard on-chip peripherals including a 16-bit central processing unit (HCS12 CPU), up to 128K bytes of Flash EEPROM or ROM, up to 6K bytes of RAM, one asynchronous serial communications interfaces (SCI), a serial peripheral interface (SPI), Clock, Reset and Power management Unit (CPMU), an IIC-bus interface (IIC), an 8-channel 16-bit timer (TIM), a 10-channel, 12-bit analog-to-digital converter (ADC), a six-channel pulse width modulator (PWM) and one multi-scalable controller area network(MSCAN).

1.2 MC9S12P Full Chip Simulation

Full Chip Simulation means not only to simulate the core instruction set but also the on chip I/O devices such as (CPMU, PWM...). In the section Simulation Summary the supported I/O devices are listed for each supported derivative of MC9S12P-Family.

HC(S) 12(X) Debugger Manual can be referred for the details of configuring HCS12 Debugger to use Full Chip Simulation connection. The Full Chip Simulation (FCS) connection runs a complete simulation of all processor peripherals and I/O on the user's Personal Computer. No development board is required. Each derivative has a totally different simulation engine to accurately simulate the memory ranges, I/O, and peripherals for a given derivative.

1.3 Scope

This document details the Full Chip Simulation support for I/O devices for the supported derivatives of MC9S12P-Family. It details the features that are simulated, not simulated, or simulated with exceptions.

1.4 References

Item	Description
MC9S12P128_Reference_Manual.pdf	MC9S12P128 Data Sheet, Covers MC9S12P Family

2 Simulation Summary

The following table gives a summary of peripheral options of MC9S12P Family members. Each derivative for P family comes with different memory, peripheral and package options. The number of peripherals for each derivative has been mentioned in the table. The modules which have been simulated are hyperlinked and details regarding them can be seen be selecting the hyperlink.

Derivative →		**MC9S 12P128**	**MC9S 12P96**	**MC9S 12P64**	**MC9S 12P32**
Peripheral Module	**Module Version**				
PIM (port integration module)	S12PPIMV1	1	1	1[6]	1[6]
MMC (memory mapping control)	S12PMMCV1[1) 2)]	1	1	1	1
DBG (debug module)	S12SDBGV2[2)]	1	1	1	1
CPMU(Clock, Reset and Power Management Unit	S12CPMUV1	1	1	1	1
Voltage regulator (Part of CPMU)	VREG	1	1	1	1
ADC (analog-to-digital converter)	ADC12B10CV1	1	1	1	1
SCI (serial communications interface)	S12SCIV5	1	1	1	1
SPI (serial peripheral interface)	S12SPIV5 [3)]	1	1	1	1
FTM control registers	S12FTMRC128K1V1[1)]	1	-	-	-
FTM control registers	S12FTMRC96K1V1[1)]	-	1	-	-
FTM control registers	S12FTMRC64K1V1[1)]	-	-	1	-
FTM control registers	S12FTMRC32K1V1[1)]	-	-	-	1
INT (interrupt module)	S12SINTV1 [4)]	1	1	1	1
CAN (controller area network)	S12MSCANV3 [3)]	1	1	1	1
PWM (pulse width modulator)	PWM8B6CV1	1	1	1	1
CPU	CPU12-V1 [4)]	1	1	1	1
TIM (timer module)	TIM16B8CV2	1	1	1	1
BDM (background debug module)	S12SBDMV1 [2) 5)]	1	1	1	1
OSC (pierce oscillator)	S12POSCLCPV1 [2) 5)]	1	1	1	1

1) Any memory is integral part of the core simulator.
2) This module is not simulated.
3) Allow read and write access but no module specific functionality.
4) This module is fully implemented and part of the core simulator.
5) This Module is not required in context of simulator.
6) Does not contain all port pins. Please check the datasheets for details.

3 Peripheral Modules – Simulation Details

3.1 Port Integration Module (S12PPIMV1)

3.1.1 Implementation of PIM module (S12PPIMV1)

Major functionalities of GPIO are implemented. GPIO functionalities like reduced output drive, wired-or functions are not implemented. Static routing and dynamic routing (routing based on routing registers) are implemented only for peripherals PWM, TIM, and CPMU. Free-running clock outputs and IRQ functionality are not implemented.

3.1.2 Extensions

PIM establishes the interface between the peripheral modules and the I/O pins for all ports. All the port pins are simulated. Routing of signals only for peripherals like PWM, TIM, and CPMU are implemented.

The ports that are available are PortA, PortB, PortE, PortT, PortS, PortM, PortP, PortJ, PortAD0 and PortAD1. Each port can be accessed by using the template mentioned below:

To Access all the bits of the port use PIM.PORT<x>Val.

To Access a particular bit of the port use PIM.PORT<x>Pin<y>.

Where x is the port name and y represents the bit position (Least significant bit is 0).

Consider the example of PORT B:

To Access all 8 bits of port B use the virtual register PIM.PORTBVal.

To Access individual bits of port B use the virtual registers below:

PIM.PORTBPin0 - Least significant bit.

PIM.PORTBPin1

PIM.PORTBPin2

PIM.PORTBPin3

PIM.PORTBPin4

PIM.PORTBPin5

PIM.PORTBPin6

PIM.PORTBPin7 - Most significant bit.

3.1.3 Restrictions

Following is the list of the registers which are not fully simulated:

All Reduced Drive Registers (RDRIV, RDRT, RDRS, RDRM, RDRP, RDRJ, RDR0AD0 and RDR1AD0) functionalities are not implemented but read/write to same has been provided.
All Wired-Or Mode Registers (WOMS, WOMM) functionalities are not implemented but read/write to same has been provided.
ECLK Control Register is not implemented and hence free running clock outputs are not provided.
Registers at the address at 0x001D, 0x001F, 0x0246, 0x0247, 0x024F, 0x036E, 0x037E are reserved for factory testing of the PIM module and are not available in normal modes.

3.1.4 Register Details

This table gives a detailed view on the registers of this peripheral module. Registers and bits not or only partially implemented will be color encoded

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 PORTA	R	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
	W								
0x0001 PORTB	R	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
	W								
0x0002 DDRA	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
	W								
0x0003 DDRB	R	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
	W								
0x0004-07 RESERVED	R	0	0	0	0	0	0	0	0
	W								
0x0008 PORTE	R	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
	W								
0x0009 DDRE	R	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	0	0
	W								
0x000A-0B RESERVED	R	0	0	0	0	0	0	0	0
	W								
0x000C PUCR	R	PUPKE	BKPUE	0	PUPEE	PUPDE	PUPCE	PUPBE	PUPAE
	W								
0x000D RDRIV	R	RDPK	0	0	RDPE	RDPD	RDPC	RDPB	RDPA
	W								

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x000E-1B	R	0	0	0	0	0	0	0	0
RESERVED	W								
0x001C	R	NECLK	NCLKX2	DIV16	EDIV4	EDIV3	EDIV2	EDIV1	EDIV0
ECLKCTL	W								
0x001D	R	0	0	0	0	0	0	0	0
RESERVED	W								
0x001E	R	IRQE	IRQEN	0	0	0	0	0	0
IRQCR	W								
0x001F	R	0	0	0	0	0	0	0	0
RESERVED	W								
0x0240	R	PTT7	PTT6	PTT5	PTT4	PTT3	PTT2	PTT1	PTT0
PTT	W								
0x0241	R	PTIT7	PTIT6	PTIT5	PTIT4	PTIT3	PTIT2	PTIT1	PTIT0
PTIT	W								
0x0242	R	DDRT7	DDRT6	DDRT5	DDRT4	DDRT3	DDRT2	DDRT1	DDRT0
DDRT	W								
0x0243	R	RDRT7	RDRT6	RDRT5	RDRT4	RDRT3	RDRT2	RDRT1	RDRT0
RDRT	W								
0x0244	R	PERT7	PERT6	PERT5	PERT4	PERT3	PERT2	PERT1	PERT0
PERT	W								
0x0245	R	PPST7	PPST6	PPST5	PPST4	PPST3	PPST2	PPST1	PPST0
PPST	W								
0x0246	R	0	0	0	0	0	0	0	0
RESERVED	W								
0x0247	R	0	0	PTTRR5	PTTRR4	0	0	0	PTTRR0
PTTRR	W								
0x0248	R	0	0	0	0	PTS3	PTS2	PTS1	PTS0
PTS	W								

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0249	R	0	0	0	0	PTIS3	PTIS2	PTIS1	PTIS0
PTIS	W								
0x024A	R	0	0	0	0	DDRS3	DDRS2	DDRS1	DDRS0
DDRS	W								
0x024B	R	0	0	0	0	RDRS3	RDRS2	RDRS1	RDRS0
RDRS	W								
0x024C	R	0	0	0	0	PERS3	PERS2	PERS1	PERS0
PERS	W								
0x024D	R	0	0	0	0	PTSP3	PTSP2	PTSP1	PTSP0
PTPS	W								
0x024E	R	0	0	0	0	WOMS3	WOMS2	WOMS1	WOMS0
WOMS	W								
0x024F	R	0	0	0	0	0	0	0	0
RESERVED	W								
0x0250	R	0	0	PTM5	PTM4	PTM3	PTM2	PTM1	PTM0
PTM	W								
0x0251	R	0	0	PTIM5	PTIM4	PTIM3	PTIM2	PTIM1	PTIM0
PTIM	W								
0x0252	R	0	0	DDRM5	DDRM4	DDRM3	DDRM2	DDRM1	DDRM0
DDRM	W								
0x0253	R	0	0	RDRM5	RDRM4	RDRM3	RDRM2	RDRM1	RDRM0
RDRM	W								
0x0254	R	0	0	PERM5	PERM4	PERM3	PERM2	PERM1	PERM0
PERM	W								
0x0255	R	0	0	PPSM5	PPSM4	PPSM3	PPSM2	PPSM1	PPSM0
PPSM	W								
0x0256	R	0	0	WOMM5	WOMM4	WOMM3	WOMM2	WOMM1	WOMM0
WOMM	W								
0x0257	R	0	0	0	0	0	0	0	0
RESERVED	W								

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0258 PTP	R	PTP7	0	PTP5	PTP4	PTP3	PTP2	PTP1	PTP0
	W								
0x0259 PTIP	R	PTIP7	0	PTIP5	PTIP4	PTIP3	PTIP2	PTIP1	PTIP0
	W								
0x025A DDRP	R	DDRP7	0	DDRP5	DDRP4	DDRP3	DDRP2	DDRP1	DDRP0
	W								
0x025B RDRP	R	RDRP7	0	RDRP5	RDRP4	RDRP3	RDRP2	RDRP1	RDRP0
	W								
0x025C PERP	R	PERP7	0	PERP5	PERP4	PERP3	PERP2	PERP1	PERP0
	W								
0x025D PTPP	R	PPSP7	0	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSP0
	W								
0x025E PIEP	R	PIEP7	0	PIEP5	PIEP4	PIEP3	PIEP2	PIEP1	PIEP0
	W								
0x025F PIFP	R	PIFP7	0	PIFP5	PIFP4	PIFP3	PIFP2	PIFP1	PIFP0
	W								
0x0260-67 RESERVED		0	0	0	0	0	0	0	0
0x0268 PTJ	R	PTJ7	PTJ6	0	0	0	PTJ2	PTJ1	PTJ0
	W								
0x0269 PTIJ	R	PTIJ7	PTIJ6	0	0	0	PTIJ2	PTIJ1	PTIJ0
	W								
0x026A DDRJ	R	DDRJ7	DDRJ6	0	0	0	DDRJ2	DDRJ1	DDRJ0
	W								
0x026B RDRJ	R	RDRJ7	RDRJ6	0	0	0	RDRJ2	RDRJ1	RDRJ0
	W								
0x026C PERJ	R	PERJ7	PERJ6	0	0	0	PERJ2	PERJ1	PERJ0
	W								
0x026D PPSJ	R	PPSJ7	PPSJ6	0	0	0	PPSJ2	PPSJ1	PPSJ0
	W								

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x026E PIEJ	R	PIEJ7	PIEJ6	0	0	0	PIEJ2	PIEJ1	PIEJ0
	W								
0x026F PIFJ	R	PIFJ7	PIFJ6	0	0	0	PIFJ2	PIFJ1	PIFJ0
	W								
0x0270 PT0AD0	R	0	0	0	0	0	0	PT0AD09	PT0AD08
	W								
0x0271 PT1AD0	R	PT1AD07	PT1AD06	PT1AD05	PT1AD04	PT1AD03	PT1AD02	PT1AD01	PT1AD00
	W								
0x0272 DDR0AD0	R	0	0	0	0	0	0	DDR0AD09	DDR0AD08
	W								
0x0273 DDR1AD0	R	DDR1AD07	DDR1AD06	DDR1AD05	DDR1AD04	DDR1AD03	DDR1AD02	DDR1AD01	DDR1AD00
	W								
0x0274 RDR0AD0	R	0	0	0	0	0	0	RDR0AD1	RDR0AD0
	W								
0x0275 RDR1AD0	R	RDR1AD07	RDR1AD06	RDR1AD05	RDR1AD04	RDR1AD03	RDR1AD02	RDR1AD01	RDR1AD00
	W								
0x0276 PER0AD0	R	0	0	0	0	0	0	PER0AD1	PER0AD0
	W								
0x0277 PER1AD0	R	PER1AD07	PER1AD06	PER1AD05	PER1AD04	PER1AD03	PER1AD02	PER1AD01	PER1AD00
	W								

= Not simulated

= Only Read / Write simulated, but no functionality

= Unimplemented or reserved (on Hardware and Simulation)

3.2 Memory Mapping Control (S12PMMCV1)

3.2.1 Implementation

The MMC module has not been fully implemented and provides only the basic functionality. The following registers have been implemented in S12PMMCV1

Direct Page Register (DIRECT)
Program Page Index Register (PPAGE)

3.2.2 Restrictions

The MMC peripheral does not provide any simulated functionality for the following set of registers neither these registers have been implemented.

MMC Control Register (MMCCTL1)

The MMCCTL1 register's IFRON bit is used to make the IFR selector visible in the global memory map.

Mode Register (MODE)

The MODE bits of the MODE register are used to establish the MCU operating mode. The external mode pins MODC determine the operating mode during RESET low (active). The state of the pins is latched into the respective register bits after the RESET signal goes inactive.

3.2.3 Register Details

This table gives a detailed view on the registers of this peripheral module. Registers and bits not or only partially implemented will be color encoded.

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x000B	R	MODC	0	0	0	0	0	0	0
MODE	W								
0x000C-0F	R	0	0	0	0	0	0	0	0
Reserved	W								
0x0011	R	DP15	DP14	DP13	DP12	DP11	DP10	DP9	DP8
DIRECT	W								
0x0012	R	0	0	0	0	0	0	0	0
Reserved	W								
0x0013	R	0	0	0	0	0	0	0	IFRON
MMCCTL1	W								

0x0014	R	0	0	0	0	0	0	0	0
Reserved	W								
0x0015	R	1	1	1	1	PIX3	PIX2	PIX1	PIX0
PPAGE	W								
0x0016	R	0	0	0	0	0	0	0	0
Reserved	W								
0x0017	R	0	0	0	0	0	0	0	0
Reserved	W								

= Not simulated

= Only Read / Write simulated, but no functionality

= Unimplemented or reserved (on Hardware and Simulation)

3.3 Clock, Reset and Power Management Unit (S12CPMUV1)

3.3.1 Implementation

The following functionalities are implemented.

RTI : Real Time Interrupt

PLL : Phase Locked Loop

COP : Computer Operating Properly watchdog

The following functionalities are not implemented.

POR : Power On Reset.

LVR : Low-Voltage Reset.

HTI : High Temperature Interrupt

The following functionality is implemented in VReg module.

API : Autonomous Periodical Interrupt

3.3.2 Extensions

None.

3.3.3 Restrictions

CPMUHTCTL, CPMULVCTL, and CPMUHTTR are the registers which are not implemented. CPMUTEST0, CPMUTEST1, CPMUTEST2 and CPMUTEST3 are the reserved registers which are designed for factory test purposes only.

Some other bits which are not simulated/not fully simulated: refer to the table below:

3.3.4 Register Details (CPMUV1)

This table gives a detailed view on the registers of this peripheral module. Registers and bits not or only partially implemented will be color encoded.

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 CPMUSYNR	R	VCOFREQ[1:0]		SYNDIV[5:0]					
	W								

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0001 CPMUREFDV	R	REFFRQ[1:0]		0	0	REFDIV[3:0]			
	W								
0x0002 CPMUREFDV	R	0	0	0	POSTDIV[4:0]				
	W								
0x0003 CPMUFLG	R	RTIF	PORF	LVRF[A)]	LOCKIF	LOCK	ILAF	OSCIF	UPOSC
	W								
0x0004 CPMUINT	R	RTIE	0	0	LOCKIE	0	0	OSCIE	0
	W								
0x0005 CPMUCLKS	R	PLLSEL	PSTP	0	0	PRE	PCE	RTIOSCSEL	COPOSCSEL
	W								
0x0006 CPMUPLL	R	0	0	FM1	FM0	0	0	0	0
	W								
0x0007 CPMURTI	R	RTDEC	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
	W								
0x0008 CPMUCOP	R	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
	W			WRTMASK					
0x0009 CPMUTEST0	R	0	0	0	0	0	0	0	0
	W								
0x000A CPMUTEST1	R	0	0	0	0	0	0	0	0
	W								
0x000B CPMUARMCOR	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
0x000A CPMUHTCTL	R	0	0	VSEL	0	HTE	HTDS	HTIE	HTIF
	W								
0x000A CPMULVCTL	R	0	0	0	0	0	LVDS	LVIE	LVIF
	W								
0x000A CPMUAPICTL[1)]	R	APICLK	0	0	APIFES	APIEA	APIFE	APIE	APIF
	W								
0x000A CPMUAPITR[1)]	R	APITR5	APITR4	APITR3	APITR2	APITR1	APITR0	0	0
	W								

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x000A CPMUAPIRH [1]	R	APIR15	APIR14	APIR13	APIR12	APIR11	APIR10	APIR9	APIR8
	W								
0x000A CPMUAPIRL [1]	R	APIR7	APIR6	APIR5	APIR4	APIR3	APIR2	APIR1	APIR0
	W								
0x000A CPMUTEST3	R	0	0	0	0	0	0	0	0
	W								
0x000B CPMUHTTR	R	HTOE	0	0	0	HTTR3	HTTR2	HTTR1	HTTR0
	W								
0x000B CPMUIRCTRIMH	R	TCTRIM[3:0]				0	0	IRCTRIM[9:8]	
	W								
0x000B CPMUIRCTRIML	R	IRCTRIM[7:0]							
	W								
0x000B CPMUOSC	R	OSCE	0	OSCFILT[5:0]					
	W								
0x000B CPMUPROT	R	0	0	0	0	0	0	0	PROT
	W								
0x000B CPMUTEST2	R	0	0	0	0	0	0	0	0
	W								

= Not simulated

= Only Read / Write simulated, but no functionality

= Unimplemented or reserved (on Hardware and Simulation)

1) API is implemented in VREG Module. All registers related to API i.e. CPMUAPICTL, CPMUAPITR, CPMUAPIRH and CPMUAPIRL are simulated in the VREG module.

3.4 Analog-to-Digital Converter (ADC12B10CV1)

3.4.1 Implementation

ADC_12B10C is implemented as an extension of ATD module separately and not over the existing ATD module implementation. Most of the features of this peripheral module have been simulated.

The basic implementation gets the total number of conversions to be done, calculates conversion time and sets up an event for each conversion.

3.4.2 Extensions

PADx

The analog inputs channels are reachable separately through the object pool. They are implemented as Non Memory Mapped ADC virtual registers called PAD0 to PAD9. For the ADC module, PAD0 input corresponds to the AN0 pin of the microcontroller. The following command is used to provide input voltage to analog channel in simulator:

ATDx_SETPAD <CHANNEL> <VOLTAGE AS FLOAT>

3.4.3 Restrictions

The unimplemented hardware features of ADC12B10CV1 are as follow:

Configurable external trigger functionality on any AD channel or any of four additional trigger inputs. The four additional trigger inputs can be chip external or internal.

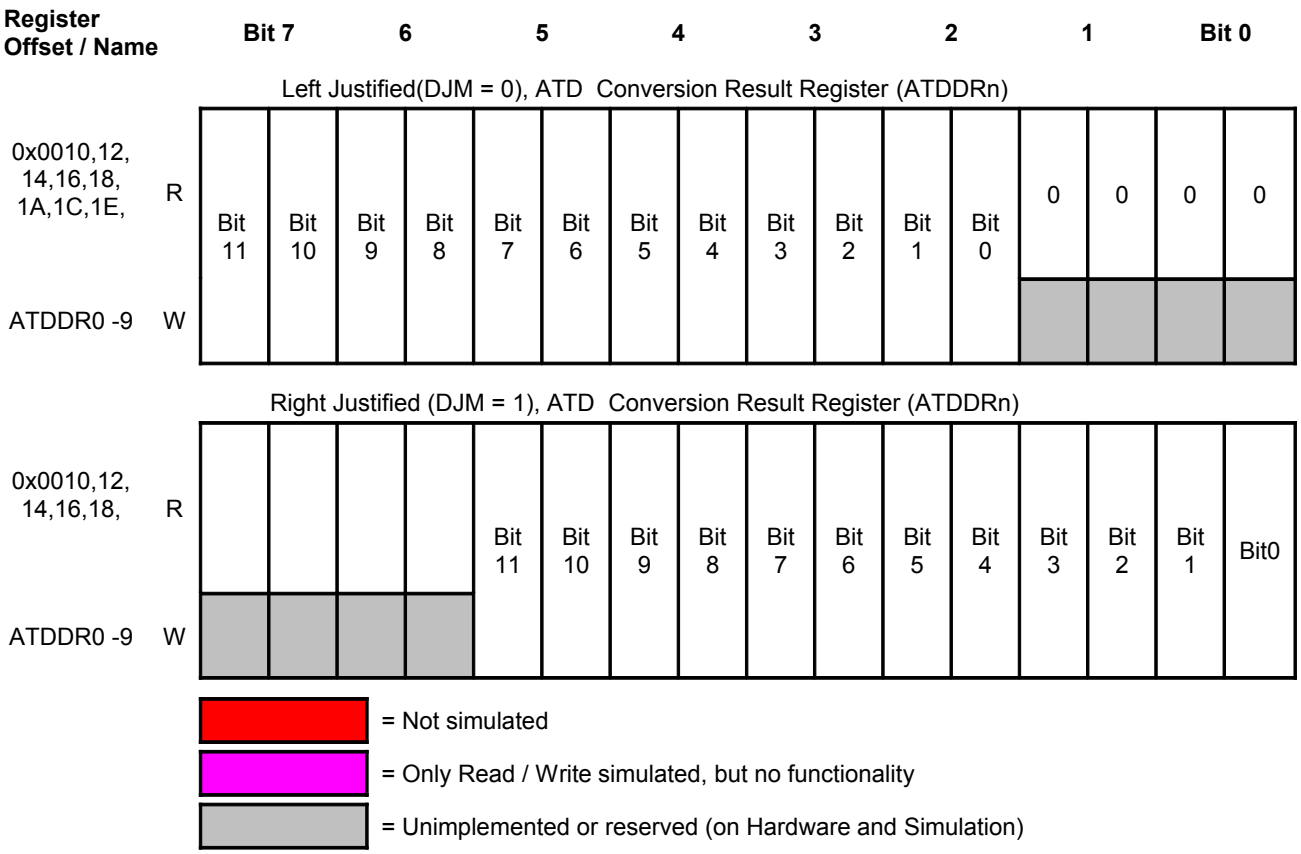
Background Debug Freeze Enable — when debugging an application, it is useful in many cases to have the ATD pause when a breakpoint (Freeze Mode) is encountered. The bits [FRZ0:1] I ATDCTL3 register determine how the ATD will respond to a breakpoint.

3.4.4 Register Details

This table gives a detailed view on the registers of this peripheral module. Registers and bits not or only partially implemented will be color encoded.

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 [ATDCTL0]	R	RESERVED	0	0	0	WRAP3	WRAP2	WRAP1	WRAP0
	W								
0x0001 [ATDCTL1]	R	ETRIGSEL	SRES1	SRES0	SMP_DIS	ETRIGCH3	ETRIGCH2	ETRIGCH1	ETRIGCH0
	W								
0x0002 [ATDCTL2]	R	0	AFFC	ICLKSTP	ETRIGLE	ETRIGP	ETRIGE	ASCIE	ACMPIE
	W								

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0003 [ATDCTL3]	R	DJM	S8C	S4C	S2C	S1C	FIFO	FRZ1	FRZ0
	W								
0x0004 [ATDCTL4]	R	SMP2	SMP1	SMP0	PRS[4:0]				
	W								
0x0005 [ATDCTL5]	R	0	SC	SCAN	MULT	CD	CC	CB	CA
	W								
0x0006 [ATDSTAT0]	R	SCF	0	ETORF	FIFOR	CC3	CC2	CC1	CC0
	W								
0x0007 Unimplemented	R	0	0	0	0	0	0	0	0
	W								
0x0008 [ATDCMPEH]	R	0	0	0	0	0	0	CMPE[9:0]	
	W								
0x0009 [ATDCMPEL]	R	CMPE[7:0]							
	W								
0x000A [ATDSTAT2H]	R	0	0	0	0	0	0	CCF[9:8]	
	W								
0x000B [ATDSTAT2L]	R	CCF[7:0]							
	W								
0x000C [ATDDIENH]	R	0	0	0	0	0	0	IEN[15:8]	
	W								
0x000D [ATDDIENL]	R	IEN[7:0]							
	W								
0x000E ATDCMPHTH	R	0	0	0	0	0	0	CMPHT[15:8]	
	W								
0x000F ATDCMPHTL	R	CMPTH[7:0]							
	W								


Register Offset / Name
Bit 7
6
5
4
3
2
1
Bit 0
Left Justified(DJM = 0), ATD Conversion Result Register (ATDDRn)
0x0010,12, 14,16,18, 1A,1C,1E,
R
ATDDR0 -9
W
Bit 11
Bit 10
Bit 9
Bit 8
Bit 7
Bit 6
Bit 5
Bit 4
Bit 3
Bit 2
Bit 1
Bit 0
0
0
0
0
Right Justified (DJM = 1), ATD Conversion Result Register (ATDDRn)
0x0010,12, 14,16,18,
R
ATDDR0 -9
W
Bit 11
Bit 10
Bit 9
Bit 8
Bit 7
Bit 6
Bit 5
Bit 4
Bit 3
Bit 2
Bit 1
Bit0
= Not simulated
= Only Read / Write simulated, but no functionality
= Unimplemented or reserved (on Hardware and Simulation)

3.5 Pulse-Width Modulator (S12PWM8B6CV1)

3.5.1 Implementation

The PWM peripheral is fully simulated.

It has six channels, each of the channels has a programmable period and duty cycle as well as a dedicated counter. A flexible clock select scheme allows a total of four different clock sources (scaled clock A/B and clock A/B) to be used with the counters. Each of the modulators can create independent continuous waveforms with software-selectable duty rates from 0% to 100%. The PWM outputs can be programmed as left aligned outputs or center aligned outputs. The channels can be six 8 bit separate or three 16 bit concatenated channels.

In emergency shutdown mode, channel 5 act as input and an interrupt is generated at the time of emergency shutdown, whenever the input level of the PWM5 channel changes. All the channels can be restarted again from the shutdown feature by setting PWM restart bit in shutdown register.

3.5.2 Extensions

The port associated with the PWM module is the port P, however its implementation has been done in the PIM module that handles communication with PWM and I/O behavior of the port when the PWM is disconnected from the pins.

The registers conditioning the connection between PWM and PIM is PWME. The virtual PORTP register in the PWM are partial images of PTP. They represent the PTP value if the PWM alone would control the port.

3.5.3 Restrictions

The PFRZ bit in register PWMCTL is not implemented . Refer to the table below.

Note: Writing the PMSDN register bit manipulating instructions MUST be avoided because it may give incorrect results. Always write whole byte in this register.

Writing to the bits of PWMSDN register as bitwise instructions, the instruction is executed like it’ll read the byte of PWMSDN register and write the bits from the bitwise instructions and write back same values in other bits. If the bit PWM Interrupt Flag(PWMIF) is 1 which reflects change on PWM5IN input, this bit is now cleared by again writing 1 on it. It may lead to incorrect results.

3.5.4 Register Details

This table gives a detailed view on the registers of this peripheral module. Registers and bits not or only partially implemented will be color encoded.

Register Offset /		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 [PWME]	R	0	0	PWME5	PWME4	PWME3	PWME2	PWME1	PWME0
	W								
0x0001 [PWMPOL]	R	0	0	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0
	W								
0x0002 [PWMCLK]	R	0	0	PCLK5	PCLK4	PCLK3	PCLK2	PCLK1	PCLK0
	W								
0x0003 [PWMPRCLK]	R	0	PCKB2	PCKB1	PCKB0	0	PCKA2	PCKA1	PCKA0
	W								
0x0004 [PWMCAE]	R	0	0	CAE5	CAE4	CAE3	CAE2	CAE1	CAE0
	W								
0x0005 [PWMCTL]	R	0	CON45	CON23	CON01	PSWAI	PFRZ	0	0
	W								
0x0006 [PWMTST]	R	0	0	0	0	0	0	0	0
	W								
0x0007 [PWMPRSC]	R	0	0	0	0	0	0	0	0
	W								
0x0008-09 PWMSCLA-B	R								
	W								
0x000A-0B PWMSCNTA-B	R	0	0	0	0	0	0	0	0
	W								
0x000C-11 [PWMCNT0-5]	R	0	0	0	0	0	0	0	0
	W								
0x0012-17 PWMPER0-5	R								
	W								
0x0018-1D PWMDTY0-7	R								
	W								
0x001E [PWMSDN]	R	PWMIF	PWMIE	0 PWMRSTRT	PWMLVL	0	PWM5IN	PWM5INL	PWM5ENA
	W								

= Not simulated

= Only Read / Write simulated, but no functionality

Register Offset /	Bit 7	6	5	4	3	2	1	Bit 0

= Unimplemented or reserved (on Hardware and Simulation)

3.6 Serial Communication Interface (S12SCIV5)

3.6.1 Implementation

The basic implementation pushes the data in the buffer and pops up the data from the buffer depending on the baud rate specified. In SCI Module the Baud Rate Generation, Transmission and Receiver functionalities have been implemented. The interrupt TDRE, TC, RDRF, OR, IDLE in SCI status register 1 (SCISR1) are supported in the SCI modules.

3.6.2 Extension

There are two non-memory mapped SCI virtual registers that are used as token interface to receive or transmit 8 or 9 bits data. The register 'SerialInput' serves to send characters to the SCI Module. The register 'SerialOutput' contain the characters sent from to the SCI Module.

SCI_DEF0_INPUT_NAME (SerialInput)

This is a non memory mapped register and its serves to connect the SCI to the terminal window. The ninth bit is not supported. A read access to SerialInput has no specified meaning. Bit 7..0 data is received from terminal window to SCI

SCI_DEF0_OUTPUT_NAME (SerialOutput)

This not memory mapped register and it serves to connect the SCI to the terminal window. The ninth bit is not supported. A write access to SerialOutput has no specified meaning. Bit 7..0 data is sent from SCI to terminal window.

3.6.3 Restrictions

In S12SCI_V5, registers SCIASR1 (0x0000), SCIACR1 (0x0001), SCIACR2 (0x00002) become visible only in the hardware and are not simulated. Also the interrupts RXEDGIF, BERRIF, BKDIF related with these registers are not supported.
The Parity check, Noise detection, SCISWAI bit and WAKE bit in SCI control register 1 (SCICR1) has not been implemented in Simulator.
Infra Red encoding and decoding support as well as Infra Red enable bit (IREN) in SCI Baud Register (SCIBDH) has not been implemented in simulator.
Noise Error Flag, Parity Flag and Framing Error Flag bits in SCI status register 2 (SCISR2) has not been implemented in simulator.
The TXDIR, BRK13, TXPOL, RXPOL and AMAP bits in SCI status register 2 (SCISR2) has not been implemented in simulator.
Lin support, Single-Wire mode and LOOP operation has not been implemented in simulator.
Before starting the test for each SCI module, TXD pin of one SCI needs to be connected to RXD pin of second SCI and vice versa using PinConn module. The same test can be run on hardware after connecting TXD and RXD pins of SCI0 and SCI1 modules i.e. only SCI0 can be connected to SCI1 and vice versa

3.6.4 Register Details

This table gives a detailed view on the registers of this peripheral module. Registers and bits not or only partially implemented will be color encoded.

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000* SCIBDH	R	IREN	TNP1	TNP0	SBR12	SBR11	SBR10	SBR9	SBR8
	W								
0x0001* SCIBDL	R	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
	W								
0x0002* SCICR1	R	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
	W								
0x0000~ SCIASR1	R	RXEDGIF	0	0	0	0	BERRV	BERRIF	BKDIF
	W								
0x0001~ SCIACR1	R	RXEDGIE	0	0	0	0	0	BERRIE	BKDIE
	W								
0x0002~ SCIACR2	R	0	0	0	0	0	BERRMI	BERRM0	BKDFE
	W								
0x0003 SCICR2	R	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
	W								
0x0004 SCISR1	R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
	W								
0x0005 SCISR2	R	AMAP	0	0	TXPOL	RXPOL	BRK13	TXDIR	RAF
	W								
0x0006 SCIDRH	R	R8	T8	0	0	0	0	0	0
	W								
0x0007 SCIDRL	R	R7	R6	R5	R4	R3	R2	R1	R0
	W	T7	T6	T5	T4	T3	T2	T1	T0

= Not simulated

= Only Read / Write simulated, but no functionality

= Unimplemented or reserved (on Hardware and Simulation)

* Those registers are accessible if the AMAP bit in the SCISR2 register is set to zero
~ Those registers are accessible if the AMAP bit in the SCISR2 register is set to one

3.7 Timer Module (TIM16B8CV2)

3.7.1 Implementation

The basic timer consists of a 16-bit, software-programmable counter driven by an enhanced programmable prescaler. The timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from microseconds to many seconds.

This version of TIM contains 8 complete input capture/output compare(IC/OC) channels and one pulse accumulator. The input capture function is used to detect a selected transition edge and record the time. The output compare function is used for generating output signals or for timer software delays. The 16-bit pulse accumulator is used to operate as a simple event counter or a gated time accumulator. The pulse accumulator shares timer channel 7 when in event mode.

3.7.2 Extensions

The port associated with the ECT and TIM module is the port T, however its implementation has been done in the PIM module that handles communication with ECT and I/O behavior of the port when the ECT /TIM is disconnected from the pins.

The registers conditioning the connection between ECT and PIM are the TIOS, and the TCTL1-4 registers. The virtual PORTT register and PORTTBit0-7 present in the ECT and TIM are partial images of PTT. They represent the PTT value if the ECT/TIM alone would control the port.

3.7.3 Restrictions

In FREEZE mode, the system behaves as if it would be in stop mode. This is due to the fact that the simulator clock stops in FREEZE mode unlike the hardware.
In the simulator, the holding registers are considered as “empty” if their corresponding IC register is “empty” as well.
The timer prescaler clock is always used as timer counter clock as the simulator handles only prescaler Clock,
The Test Mode which is one of the special mode of operation is not available in simulator
The Pulse Accumulator A (PAA) does not the simulate the clock feedback feature to the timer entry.

3.7.4 Register Details

This table gives a detailed view on the registers of this peripheral module. Registers and bits not or only partially implemented will be color encoded.

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 TIOS	R W	IOS7	IOS6	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0001 CFORC	R	0	0	0	0	0	0	0	0
	W	FOC7	FOC6	FOC5	FOC4	FOC3	FOC2	FOC1	FOC0
0x0002 OC7M	R	OC7M7	OC7M6	OC7M5	OC7M4	OC7M3	OC7M2	OC7M1	OC7M0
	W								
0x0003 OC7D	R	OC7D7	OC7D6	OC7D5	OC7D4	OC7D3	OC7D2	OC7D1	OC7D0
	W								
0x0004 TCNTH	R	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9	TCNT8
	W								
0x0005 TCNTL	R	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0
	W								
0x0006 TSCR1	R	TEN	TSWAI	TSFRZ	TFFCA	PRNT	0	0	0
	W								
0x0007 TTOV	R	TOV7	TOV6	TOV5	TOV4	TOV3	TOV2	TOV1	TOV0
	W								
0x0008 TCTL1	R	OM7	OM7	OM6	OM6	OM5	OL5	OM4	OL4
	W								
0x0009 TCTL2	R	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
	W								
0x000A TCTL3	R	EDG7B	EDG7A	EDG6B	EDG6A	EDG5B	EDG5A	EDG4B	EDG4A
	W								
0x000B TCTL4	R	EDG1B	EDG0A	EDG1B	EDG0A	EDG1B	EDG0A	EDG0B	EDG0A
	W								
0x000C TIE	R	C7I	C6I	C5I	C4I	C3I	C2I	C1I	C0I
	W								
0x000D TSCR2	R	TOI	0	0	0	TCRE	PR2	PR1	PR0
	W								
0x000E TFLG1	R	C7F	C6F	C5F	C4F	C3F	C2F	C1F	C0F
	W								
0x000F TFLG2	R	TOF	0	0	0	0	0	0	0
	W								

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0010 - 0x001F	R	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	W								
TCxH -TCxL	R	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	W								
0x0020	R	0	PAEN	PAMOD	PEDGE	CLK1	CLK0	PAOVI	PAI
PACTL	W								
0x021	R	0	0	0	0	0	0	PAOVF	PAIF
PAFLG	W								
0x0022	R	PACNT15	PACNT14	PACNT13	PACNT12	PACNT11	PACNT10	PACNT9	PACNT8
PACNTH	W								
0x0023	R	PACNT7	PACNT6	PACNT5	PACNT4	PACNT3	PACNT2	PACNT1	PACNT0
PACNTL	W								
0x0024 - 2B	R								
RESERVED	W								
0x002C	R	OCPD7	OCPD6	OCPD5	OCPD4	OCPD3	OCPD2	OCPD1	OCPD0
OCPD	W								
0x002D	R	PTPS7	PTPS6	PTPS5	PTPS4	PTPS3	PTPS2	PTPS1	PTPS0
PTPSR	W								
0x002E	R								
RESERVED	W								

= Not simulated

= Only Read / Write simulated, but no functionality

= Unimplemented or reserved (on Hardware and Simulation)

3.8 Voltage Regulator (CPMUV1)

3.8.1 Implementation

In the VREG module, only autonomous periodical interrupt (API) functionality is implemented. When internal RC oscillator is used as clock source, the period can be increased or decreased by 25% using trimming register.

3.8.2 Extensions

The virtual register APIout is implemented which is used to generate a clock or a high pulse at the end of a selected period, depending on the configuration of APIES bit in CPMUAPICTL register. This signal is routed to a port pin.

3.8.3 Restrictions

None of the Voltage regulators are implemented.

Following modes are not implemented.

Reduced power mode (RPM) (MCU is in stop mode)
Shutdown mode

3.8.4 Register Details

This table gives a detailed view on the registers of this peripheral module. Registers and bits not or only partially implemented will be color encoded.

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x02F0 CPMUHTCL	R	0	0	VSEL	0	HTE	HTDS	HTIE	HTIF
	W								
0x02F1 CPMULVCTR	R	0	0	0	0	0	LVDS	LVIE	LVIF
	W								
0x02F2 CPMUAPICTL	R	APICLK	0	0	APIES	APIEA	APIFE	APIE	APIF
	W								
0x02F3 CPMUAPITR	R	APITR5	APITR4	APITR3	APITR2	APITR1	APITR0	0	0
	W								
0x02F4 CPMUAPIRH	R	APIR15	APIR14	APIR13	APIR12	APIR11	APIR10	APIR9	APIR8
	W								

Register Offset / Name		Bit 7	6	5	4	3	2	1	Bit 0
0x02F5 CPMUAPIRL	R	APIR7	APIR6	APIR5	APIR4	APIR3	APIR2	APIR1	APIR0
	W								
0x02F6 RESERVED	R	0	0	0	0	0	0	0	0
	W								
0x02F7 CPMUHTTR	R	HTOE	0	0	0	HTTR3	HTTR2	HTTR1	HTTR0
	W								

= Not simulated

= Only Read / Write simulated, but no functionality

= Unimplemented or reserved (on Hardware and Simulation)