



Welcome to E-XFL.COM

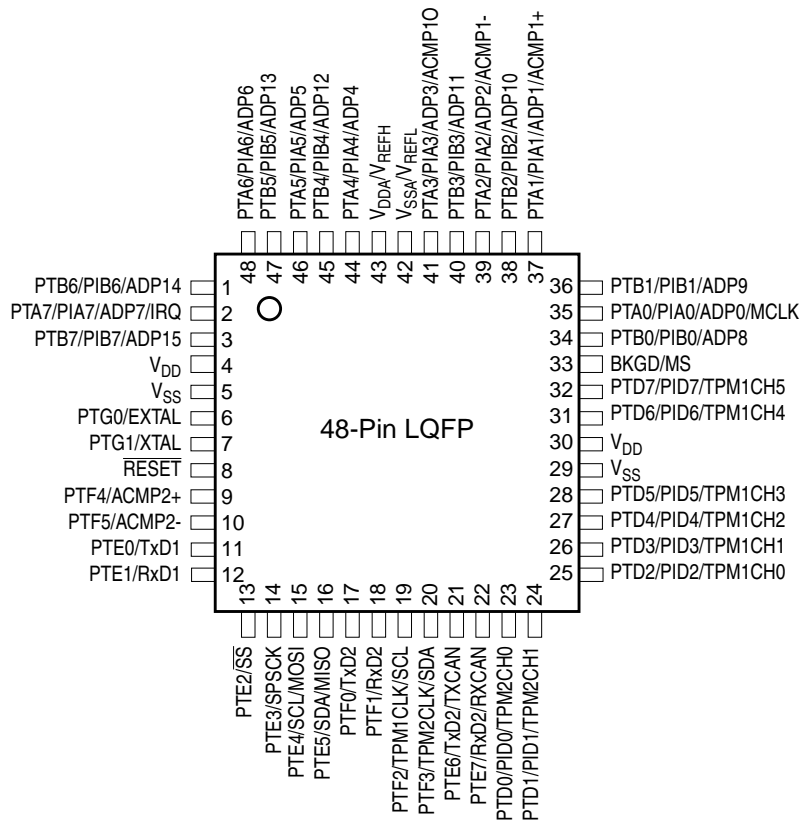
What is "[Embedded - Microcontrollers](#)"?

"[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	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I ² C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s08dv60amlh



V_{REFH} and V_{REFL} are internally connected to V_{DDA} and V_{SSA}, respectively.

Figure 2-2. 48-Pin LQFP

6.5.7 Port G Registers

Port G is controlled by the registers listed below.

6.5.7.1 Port G Data Register (PTGD)

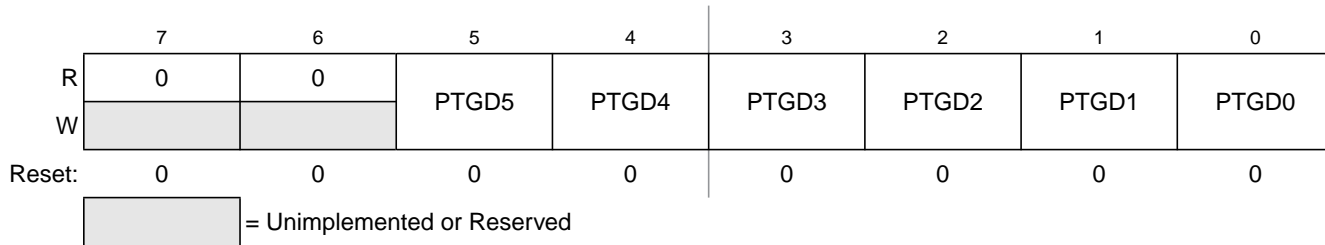


Figure 6-42. Port G Data Register (PTGD)

Table 6-40. PTGD Register Field Descriptions

Field	Description
5:0 PTGD[5:0]	Port G Data Register Bits — For port G pins that are inputs, reads return the logic level on the pin. For port G pins that are configured as outputs, reads return the last value written to this register. Writes are latched into all bits of this register. For port G pins that are configured as outputs, the logic level is driven out the corresponding MCU pin. Reset forces PTGD to all 0s, but these 0s are not driven out the corresponding pins because reset also configures all port pins as high-impedance inputs with pull-ups disabled.

6.5.7.2 Port G Data Direction Register (PTGDD)

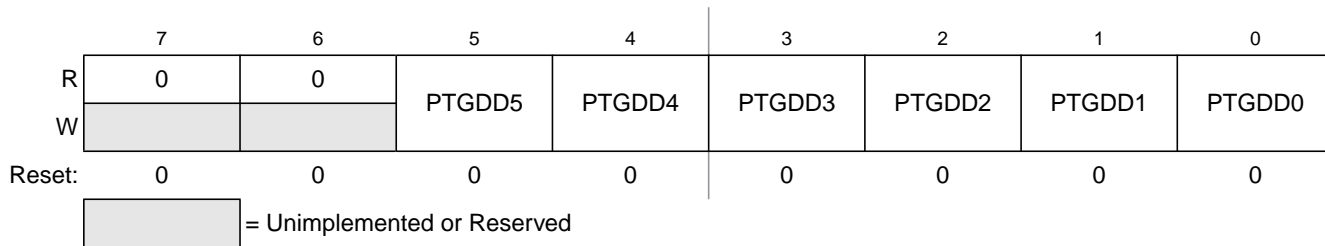


Figure 6-43. Port G Data Direction Register (PTGDD)

Table 6-41. PTGDD Register Field Descriptions

Field	Description
5:0 PTGDD[5:0]	Data Direction for Port G Bits — These read/write bits control the direction of port G pins and what is read for PTGD reads. 0 Input (output driver disabled) and reads return the pin value. 1 Output driver enabled for port G bit n and PTGD reads return the contents of PTGDn.

6.5.7.3 Port G Pull Enable Register (PTGPE)

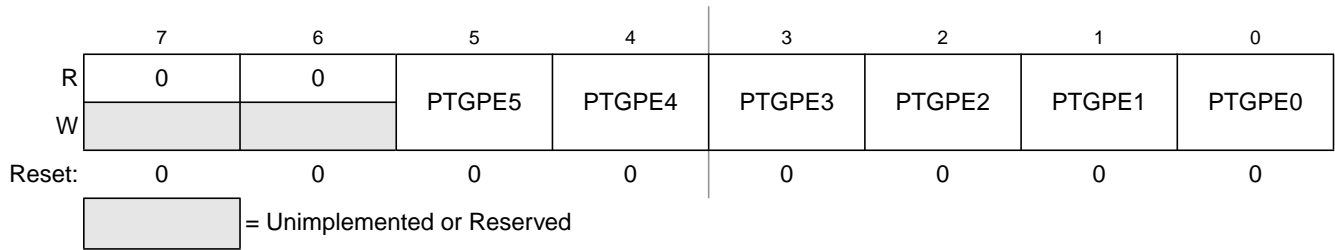


Figure 6-44. Internal Pull Enable for Port G Register (PTGPE)

Table 6-42. PTGPE Register Field Descriptions

Field	Description
5:0 PTGPE[5:0]	Internal Pull Enable for Port G Bits — Each of these control bits determines if the internal pull-up device is enabled for the associated PTG pin. For port G pins that are configured as outputs, these bits have no effect and the internal pull devices are disabled. 0 Internal pull-up device disabled for port G bit n. 1 Internal pull-up device enabled for port G bit n.

NOTE

Pull-down devices only apply when using pin interrupt functions, when corresponding edge select and pin select functions are configured.

6.5.7.4 Port G Slew Rate Enable Register (PTGSE)

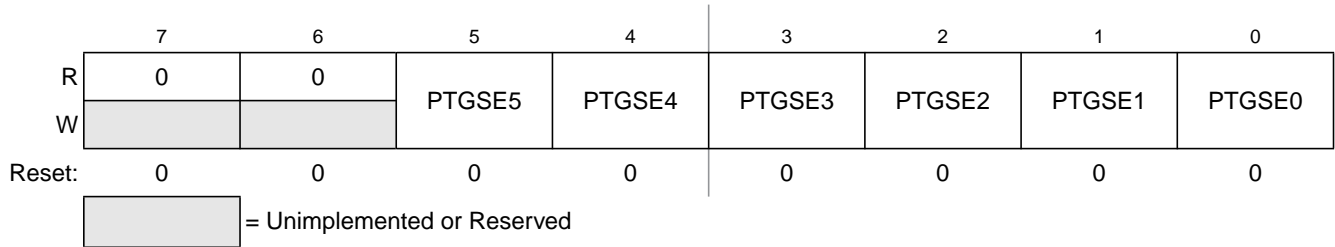


Figure 6-45. Slew Rate Enable for Port G Register (PTGSE)

Table 6-43. PTGSE Register Field Descriptions

Field	Description
5:0 PTGSE[5:0]	Output Slew Rate Enable for Port G Bits — Each of these control bits determines if the output slew rate control is enabled for the associated PTG pin. For port G pins that are configured as inputs, these bits have no effect. 0 Output slew rate control disabled for port G bit n. 1 Output slew rate control enabled for port G bit n.

Note: Slew rate reset default values may differ between engineering samples and final production parts. Always initialize slew rate control to the desired value to ensure correct operation.

Table 7-2. Instruction Set Summary (Sheet 6 of 9)

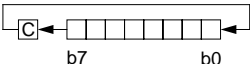
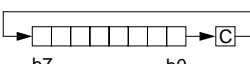
Source Form	Operation	Address Mode	Object Code	Cycles	Cyc-by-Cyc Details	Affect on CCR	
						V 1 1 H	I N Z C
MOV <i>opr8a,opr8a</i> MOV <i>opr8a,X+</i> MOV <i>#opr8i,opr8a</i> MOV <i>,X+,opr8a</i>	Move $(M)_{\text{destination}} \leftarrow (M)_{\text{source}}$ In IX+/DIR and DIR/IX+ Modes, $H:X \leftarrow (H:X) + \$0001$	DIR/DIR DIR/IX+ IMM/DIR IX+/DIR	4E dd dd 5E dd 6E ii dd 7E dd	5 5 4 5	rwppp rwwpp pwpp rwwpp	0 1 1 -	- \uparrow \downarrow -
MUL	Unsigned multiply $X:A \leftarrow (X) \times (A)$	INH	42	5	ffffp	- 1 1 0	- - - 0
NEG <i>opr8a</i> NEGA NEGX NEG <i>opr8,X</i> NEG <i>,X</i> NEG <i>opr8,SP</i>	Negate $M \leftarrow -(M) = \$00 - (M)$ (Two's Complement) $A \leftarrow -(A) = \$00 - (A)$ $X \leftarrow -(X) = \$00 - (X)$ $M \leftarrow -(M) = \$00 - (M)$ $M \leftarrow -(M) = \$00 - (M)$ $M \leftarrow -(M) = \$00 - (M)$	DIR INH INH IX1 IX SP1	30 dd 40 50 60 ff 70 9E 60 ff	5 1 1 5 4 6	rwwpp p p rwwpp rwwp prwwpp	\uparrow 1 1 -	- \uparrow \downarrow \uparrow \downarrow
NOP	No Operation — Uses 1 Bus Cycle	INH	9D	1	p	- 1 1 -	- - - -
NSA	Nibble Swap Accumulator $A \leftarrow (A[3:0]:A[7:4])$	INH	62	1	p	- 1 1 -	- - - -
ORA <i>#opr8i</i> ORA <i>opr8a</i> ORA <i>opr16a</i> ORA <i>opr8,X</i> ORA <i>opr8,X</i> ORA <i>,X</i> ORA <i>opr8,SP</i> ORA <i>opr8,SP</i>	Inclusive OR Accumulator and Memory $A \leftarrow (A) (M)$	IMM DIR EXT IX2 IX1 IX SP2 SP1	AA ii BA dd CA hh ll DA ee ff EA ff FA 9E DA ee ff 9E EA ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rwp pprpp prpp	0 1 1 -	- \uparrow \downarrow -
PSHA	Push Accumulator onto Stack $\text{Push } (A); SP \leftarrow (SP) - \0001	INH	87	2	sp	- 1 1 -	- - - -
PSHH	Push H (Index Register High) onto Stack $\text{Push } (H); SP \leftarrow (SP) - \0001	INH	8B	2	sp	- 1 1 -	- - - -
PSHX	Push X (Index Register Low) onto Stack $\text{Push } (X); SP \leftarrow (SP) - \0001	INH	89	2	sp	- 1 1 -	- - - -
PULA	Pull Accumulator from Stack $SP \leftarrow (SP + \$0001); \text{Pull } (A)$	INH	86	3	ufp	- 1 1 -	- - - -
PULH	Pull H (Index Register High) from Stack $SP \leftarrow (SP + \$0001); \text{Pull } (H)$	INH	8A	3	ufp	- 1 1 -	- - - -
PULX	Pull X (Index Register Low) from Stack $SP \leftarrow (SP + \$0001); \text{Pull } (X)$	INH	88	3	ufp	- 1 1 -	- - - -
ROL <i>opr8a</i> ROLA ROLX ROL <i>opr8,X</i> ROL <i>,X</i> ROL <i>opr8,SP</i>	Rotate Left through Carry 	DIR INH INH IX1 IX SP1	39 dd 49 59 69 ff 79 9E 69 ff	5 1 1 5 4 6	rwwpp p p rwwpp rwwp prwwpp	\uparrow 1 1 -	- \uparrow \downarrow \uparrow \downarrow
ROR <i>opr8a</i> RORA RORX ROR <i>opr8,X</i> ROR <i>,X</i> ROR <i>opr8,SP</i>	Rotate Right through Carry 	DIR INH INH IX1 IX SP1	36 dd 46 56 66 ff 76 9E 66 ff	5 1 1 5 4 6	rwwpp p p rwwpp rwwp prwwpp	\uparrow 1 1 -	- \uparrow \downarrow \uparrow \downarrow

Table 7-3. Opcode Map (Sheet 1 of 2)

Bit-Manipulation			Branch		Read-Modify-Write										Control			Register/Memory													
00	5	10	5	20	3	30	5	40	1	50	1	60	5	70	4	80	9	90	3	A0	2	B0	3	C0	4	D0	4	E0	3	F0	3
BRSET0	DIR	BSET0	DIR	BRA	REL	NEG	DIR	NEGA	INH	NEGX	INH	NEG	IX1	NEG	IX	RTI	INH	BGE	REL	SUB	IMM	SUB	DIR	SUB	EXT	SUB	IX2	SUB	IX1	SUB	IX
01	5	11	5	21	3	31	5	41	4	51	4	61	5	71	5	81	6	91	3	A1	2	B1	3	C1	4	D1	4	E1	3	F1	3
BRCLR0	DIR	BCLR0	DIR	BRN	REL	CBEQ	DIR	CBEQA	IMM	CBEQX	IMM	CBEQ	IX1+	CBEQ	IX+	RTS	INH	BLT	REL	CMP	IMM	CMP	DIR	CMP	EXT	CMP	IX2	CMP	IX1	CMP	IX
02	5	12	5	22	3	32	5	42	5	52	6	62	1	72	1	82	5+	92	3	A2	2	B2	3	C2	4	D2	4	E2	3	F2	3
BRSET1	DIR	BSET1	DIR	BHI	REL	LDHX	EXT	MUL	INH	DIV	INH	NSA	INH	DAA	INH	BGND	INH	BGT	REL	SBC	IMM	SBC	DIR	SBC	EXT	SBC	IX2	SBC	IX1	SBC	IX
03	5	13	5	23	3	33	5	43	1	53	1	63	5	73	4	83	11	93	3	A3	2	B3	3	C3	4	D3	4	E3	3	F3	3
BRCLR1	DIR	BCLR1	DIR	BLS	REL	COM	DIR	COMA	INH	COMX	INH	COM	IX1	COM	IX	SWI	INH	BLE	REL	CPX	IMM	CPX	DIR	CPX	EXT	CPX	IX2	CPX	IX1	CPX	IX
04	5	14	5	24	3	34	5	44	1	54	1	64	5	74	4	84	1	94	2	A4	2	B4	3	C4	4	D4	4	E4	3	F4	3
BRSET2	DIR	BSET2	DIR	BCC	REL	LSR	DIR	LSRA	INH	LSRX	INH	LSR	IX1	LSR	IX	TAP	INH	TXS	INH	AND	IMM	AND	DIR	AND	EXT	AND	IX2	AND	IX1	AND	IX
05	5	15	5	25	3	35	4	45	3	55	4	65	3	75	5	85	1	95	2	A5	2	B5	3	C5	4	D5	4	E5	3	F5	3
BRCLR2	DIR	BCLR2	DIR	BCS	REL	STHX	DIR	LDHX	IMM	LDHX	DIR	CPHX	IMM	CPHX	DIR	TPA	INH	TSX	INH	BIT	IMM	BIT	DIR	BIT	EXT	BIT	IX2	BIT	IX1	BIT	IX
06	5	16	5	26	3	36	5	46	1	56	1	66	5	76	4	86	3	96	5	A6	2	B6	3	C6	4	D6	4	E6	3	F6	3
BRSET3	DIR	BSET3	DIR	BNE	REL	ROR	DIR	RORA	INH	RORX	INH	ROR	IX1	ROR	IX	PULA	INH	STHX	EXT	LDA	IMM	LDA	DIR	LDA	EXT	LDA	IX2	LDA	IX1	LDA	IX
07	5	17	5	27	3	37	5	47	1	57	1	67	5	77	4	87	2	97	1	A7	2	B7	3	C7	4	D7	4	E7	3	F7	2
BRCLR3	DIR	BCLR3	DIR	BEQ	REL	ASR	DIR	ASRA	INH	ASRX	INH	ASR	IX1	ASR	IX	PSHA	INH	TAX	INH	AIS	IMM	STA	DIR	STA	EXT	STA	IX2	STA	IX1	STA	IX
08	5	18	5	28	3	38	5	48	1	58	1	68	5	78	4	88	3	98	1	A8	2	B8	3	C8	4	D8	4	E8	3	F8	3
BRSET4	DIR	BSET4	DIR	BHCC	REL	LSL	DIR	LSLA	INH	LSLX	INH	LSL	IX1	LSL	IX	PULX	INH	CLC	INH	EOR	IMM	EOR	DIR	EOR	EXT	EOR	IX2	EOR	IX1	EOR	IX
09	5	19	5	29	3	39	5	49	1	59	1	69	5	79	4	89	2	99	1	A9	2	B9	3	C9	4	D9	4	E9	3	F9	3
BRCLR4	DIR	BCLR4	DIR	BHCS	REL	ROL	DIR	ROLA	INH	ROLX	INH	ROL	IX1	ROL	IX	PSHX	INH	SEC	INH	ADC	IMM	ADC	DIR	ADC	EXT	ADC	IX2	ADC	IX1	ADC	IX
0A	5	1A	5	2A	3	3A	5	4A	1	5A	1	6A	5	7A	4	8A	3	9A	1	AA	2	BA	3	CA	4	DA	4	EA	3	FA	3
BRSET5	DIR	BSET5	DIR	BPL	REL	DEC	DIR	DECA	INH	DECX	INH	DEC	IX1	DEC	IX	PULH	INH	CLI	INH	ORA	IMM	ORA	DIR	ORA	EXT	ORA	IX2	ORA	IX1	ORA	IX
0B	5	1B	5	2B	3	3B	7	4B	4	5B	4	6B	7	7B	6	8B	2	9B	1	AB	2	BB	3	CB	4	DB	4	EB	3	FB	3
BRCLR5	DIR	BCLR5	DIR	BMI	REL	DBNZ	DIR	DBNZA	INH	DBNZX	INH	DBNZ	IX1	DBNZ	IX	PSHH	INH	SEI	INH	ADD	IMM	ADD	DIR	ADD	EXT	ADD	IX2	ADD	IX1	ADD	IX
0C	5	1C	5	2C	3	3C	5	4C	1	5C	1	6C	5	7C	4	8C	1	9C	1			BC	3	CC	4	DC	4	EC	3	FC	3
BRSET6	DIR	BSET6	DIR	BMC	REL	INC	DIR	INCA	INH	INCX	INH	INC	IX1	INC	IX	CLRH	INH	RSP	INH			JMP	DIR	JMP	EXT	JMP	IX2	JMP	IX1	JMP	IX
0D	5	1D	5	2D	3	3D	4	4D	1	5D	1	6D	4	7D	3			9D	1	AD	5	BD	5	CD	6	DD	6	ED	5	FD	5
BRCLR6	DIR	BCLR6	DIR	BMS	REL	TST	DIR	TSTA	INH	TSTX	INH	TST	IX1	TST	IX			NOP	INH	BSR	REL	JSR	DIR	JSR	EXT	JSR	IX2	JSR	IX1	JSR	IX
0E	5	1E	5	2E	3	3E	6	4E	5	5E	5	6E	4	7E	5	8E	2+	9E	Page 2	AE	2	BE	3	CE	4	DE	4	EE	3	FE	3
BRSET7	DIR	BSET7	DIR	BIL	REL	CPHX	EXT	MOV	DD	MOV	DIX+	MOV	IMD	MOV	IX+D	STOP	INH			LDX	IMM	LDX	DIR	LDX	EXT	LDX	IX2	LDX	IX1	LDX	IX
0F	5	1F	5	2F	3	3F	5	4F	1	5F	1	6F	5	7F	4	8F	2+	9F	1	AF	2	BF	3	CF	4	DF	4	EF	3	FF	2
BRCLR7	DIR	BCLR7	DIR	BIH	REL	CLR	DIR	CLRA	INH	CLR	INH	CLR	IX1	CLR	IX	WAIT	INH	TXA	INH	AIX	IMM	STX	DIR	STX	EXT	STX	IX2	STX	IX1	STX	IX

INH Inherent
 IMM Immediate
 DIR Direct
 EXT Extended
 DD DIR to DIR
 IX+D IX+ to DIR
 REL Relative
 IX Indexed, No Offset
 IX1 Indexed, 8-Bit Offset
 IX2 Indexed, 16-Bit Offset
 IMM to DIR
 DIR to IX+
 SP1 Stack Pointer, 8-Bit Offset
 SP2 Stack Pointer, 16-Bit Offset
 IX+ Indexed, No Offset with Post Increment
 IX1+ Indexed, 1-Byte Offset with Post Increment

Opcode in Hexadecimal F0 SUB 3
 Number of Bytes 1 SUB IX
 HCS08 Cycles Instruction Mnemonic Addressing Mode

The table below shows MCGOUT frequency calculations using RDIV, BDIV, and VDIV settings for each clock mode. The bus frequency is equal to MCGOUT divided by 2.

Table 8-6. MCGOUT Frequency Calculation Options

Clock Mode	f_{MCGOUT}^1	Note
FEI (FLL engaged internal)	$(f_{\text{int}} * 1024) / B$	Typical $f_{\text{MCGOUT}} = 16$ MHz immediately after reset. RDIV bits set to %000.
FEE (FLL engaged external)	$(f_{\text{ext}} / R * 1024) / B$	f_{ext} / R must be in the range of 31.25 kHz to 39.0625 kHz
FBE (FLL bypassed external)	f_{ext} / B	f_{ext} / R must be in the range of 31.25 kHz to 39.0625 kHz
FBI (FLL bypassed internal)	f_{int} / B	Typical $f_{\text{int}} = 32$ kHz
PEE (PLL engaged external)	$[(f_{\text{ext}} / R) * M] / B$	f_{ext} / R must be in the range of 1 MHz to 2 MHz
PBE (PLL bypassed external)	f_{ext} / B	f_{ext} / R must be in the range of 1 MHz to 2 MHz
BLPI (Bypassed low power internal)	f_{int} / B	
BLPE (Bypassed low power external)	f_{ext} / B	

¹ R is the reference divider selected by the RDIV bits, B is the bus frequency divider selected by the BDIV bits, and M is the multiplier selected by the VDIV bits.

This section will include 3 mode switching examples using a 4 MHz external crystal. If using an external clock source less than 1 MHz, the MCG should not be configured for any of the PLL modes (PEE and PBE).

8.5.2.1 Example # 1: Moving from FEI to PEE Mode: External Crystal = 4 MHz, Bus Frequency = 8 MHz

In this example, the MCG will move through the proper operational modes from FEI to PEE mode until the 4 MHz crystal reference frequency is set to achieve a bus frequency of 8 MHz. Because the MCG is in FEI mode out of reset, this example also shows how to initialize the MCG for PEE mode out of reset. First, the code sequence will be described. Then a flowchart will be included which illustrates the sequence.

1. First, FEI must transition to FBE mode:
 - a) MCGC2 = 0x36 (%00110110)
 - BDIV (bits 7 and 6) set to %00, or divide-by-1
 - RANGE (bit 5) set to 1 because the frequency of 4 MHz is within the high frequency range
 - HGO (bit 4) set to 1 to configure external oscillator for high gain operation
 - EREFS (bit 2) set to 1, because a crystal is being used
 - ERCLKEN (bit 1) set to 1 to ensure the external reference clock is active
 - b) Loop until OSCINIT (bit 1) in MCGSC is 1, indicating the crystal selected by the EREFS bit has been initialized.

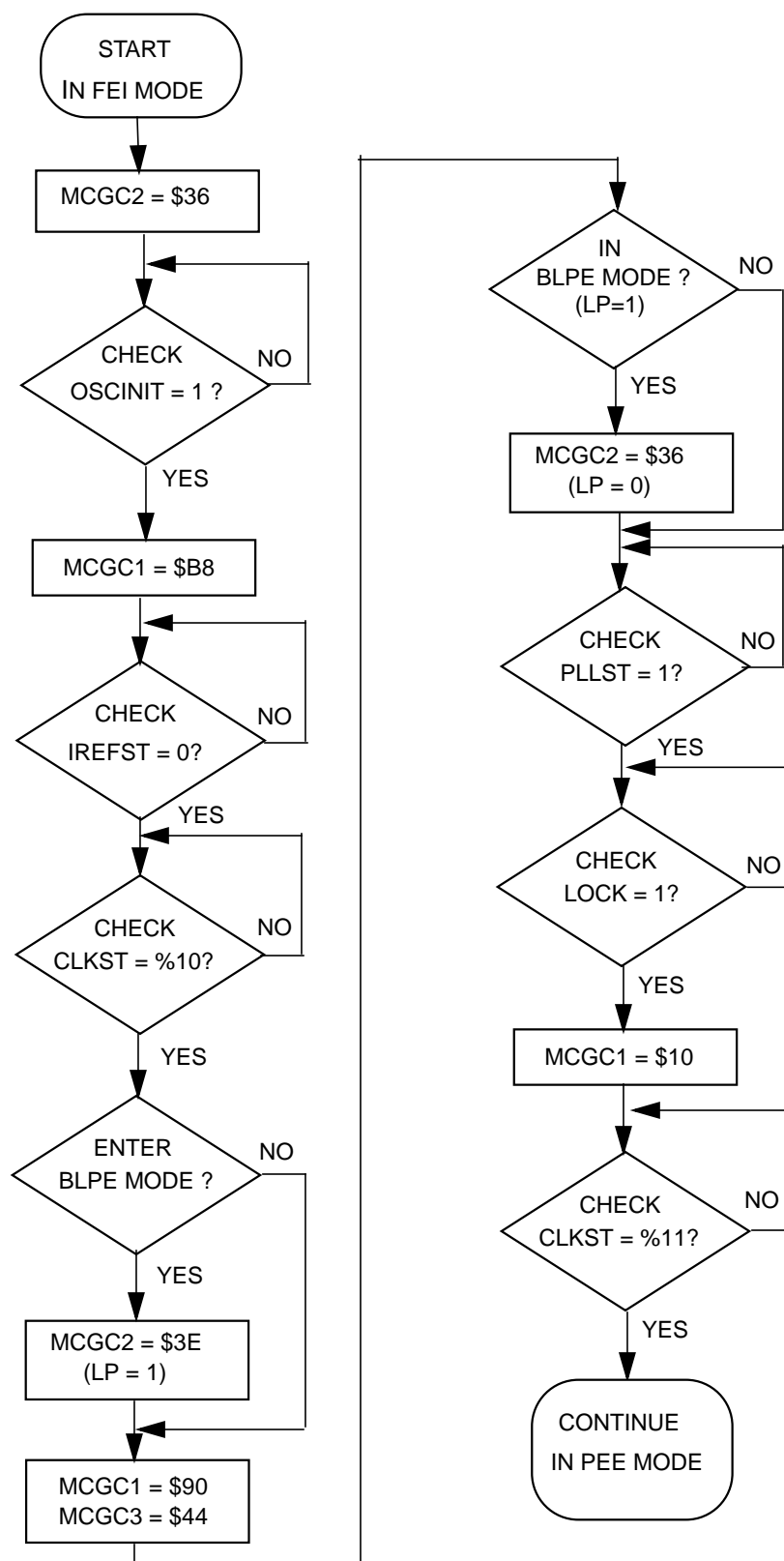


Figure 8-9. Flowchart of FEI to PEE Mode Transition using a 4 MHz crystal

used to control the pins associated with channels 0–7 of the ADC module.

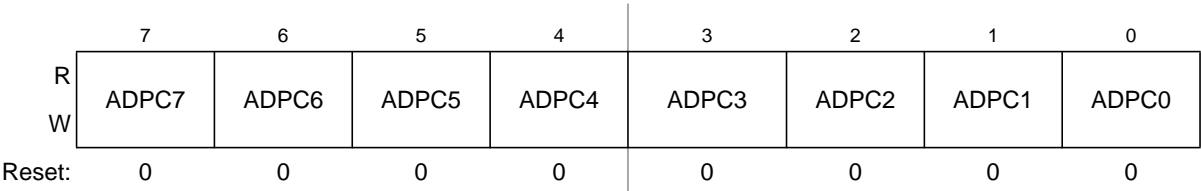


Figure 10-10. Pin Control 1 Register (APCTL1)

Table 10-10. APCTL1 Register Field Descriptions

Field	Description
7 ADPC7	ADC Pin Control 7. ADPC7 controls the pin associated with channel AD7. 0 AD7 pin I/O control enabled 1 AD7 pin I/O control disabled
6 ADPC6	ADC Pin Control 6. ADPC6 controls the pin associated with channel AD6. 0 AD6 pin I/O control enabled 1 AD6 pin I/O control disabled
5 ADPC5	ADC Pin Control 5. ADPC5 controls the pin associated with channel AD5. 0 AD5 pin I/O control enabled 1 AD5 pin I/O control disabled
4 ADPC4	ADC Pin Control 4. ADPC4 controls the pin associated with channel AD4. 0 AD4 pin I/O control enabled 1 AD4 pin I/O control disabled
3 ADPC3	ADC Pin Control 3. ADPC3 controls the pin associated with channel AD3. 0 AD3 pin I/O control enabled 1 AD3 pin I/O control disabled
2 ADPC2	ADC Pin Control 2. ADPC2 controls the pin associated with channel AD2. 0 AD2 pin I/O control enabled 1 AD2 pin I/O control disabled
1 ADPC1	ADC Pin Control 1. ADPC1 controls the pin associated with channel AD1. 0 AD1 pin I/O control enabled 1 AD1 pin I/O control disabled
0 ADPC0	ADC Pin Control 0. ADPC0 controls the pin associated with channel AD0. 0 AD0 pin I/O control enabled 1 AD0 pin I/O control disabled

10.3.9 Pin Control 2 Register (APCTL2)

APCTL2 controls channels 8–15 of the ADC module.

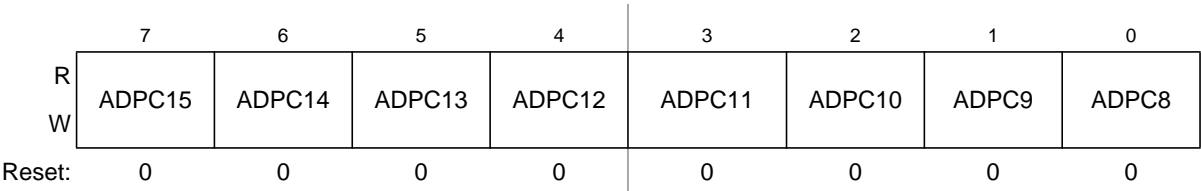


Figure 10-11. Pin Control 2 Register (APCTL2)

10.4.1 Clock Select and Divide Control

One of four clock sources can be selected as the clock source for the ADC module. This clock source is then divided by a configurable value to generate the input clock to the converter (ADCK). The clock is selected from one of the following sources by means of the ADICLK bits.

- The bus clock, which is equal to the frequency at which software is executed. This is the default selection following reset.
- The bus clock divided by two. For higher bus clock rates, this allows a maximum divide by 16 of the bus clock.
- ALTCLK, as defined for this MCU (See module section introduction).
- The asynchronous clock (ADACK). This clock is generated from a clock source within the ADC module. When selected as the clock source, this clock remains active while the MCU is in wait or stop3 mode and allows conversions in these modes for lower noise operation.

Whichever clock is selected, its frequency must fall within the specified frequency range for ADCK. If the available clocks are too slow, the ADC do not perform according to specifications. If the available clocks are too fast, the clock must be divided to the appropriate frequency. This divider is specified by the ADIV bits and can be divide-by 1, 2, 4, or 8.

10.4.2 Input Select and Pin Control

The pin control registers (APCTL3, APCTL2, and APCTL1) disable the I/O port control of the pins used as analog inputs. When a pin control register bit is set, the following conditions are forced for the associated MCU pin:

- The output buffer is forced to its high impedance state.
- The input buffer is disabled. A read of the I/O port returns a zero for any pin with its input buffer disabled.
- The pullup is disabled.

10.4.3 Hardware Trigger

The ADC module has a selectable asynchronous hardware conversion trigger, ADHWT, that is enabled when the ADTRG bit is set. This source is not available on all MCUs. Consult the module introduction for information on the ADHWT source specific to this MCU.

When ADHWT source is available and hardware trigger is enabled (ADTRG=1), a conversion is initiated on the rising edge of ADHWT. If a conversion is in progress when a rising edge occurs, the rising edge is ignored. In continuous convert configuration, only the initial rising edge to launch continuous conversions is observed. The hardware trigger function operates in conjunction with any of the conversion modes and configurations.

10.4.4 Conversion Control

Conversions can be performed in 12-bit mode, 10-bit mode, or 8-bit mode as determined by the MODE bits. Conversions can be initiated by a software or hardware trigger. In addition, the ADC module can be

10.6.2.3 Noise-Induced Errors

System noise that occurs during the sample or conversion process can affect the accuracy of the conversion. The ADC accuracy numbers are guaranteed as specified only if the following conditions are met:

- There is a 0.1 μF low-ESR capacitor from V_{REFH} to V_{REFL} .
- There is a 0.1 μF low-ESR capacitor from V_{DDAD} to V_{SSAD} .
- If inductive isolation is used from the primary supply, an additional 1 μF capacitor is placed from V_{DDAD} to V_{SSAD} .
- V_{SSAD} (and V_{REFL} , if connected) is connected to V_{SS} at a quiet point in the ground plane.
- Operate the MCU in wait or stop3 mode before initiating (hardware triggered conversions) or immediately after initiating (hardware or software triggered conversions) the ADC conversion.
 - For software triggered conversions, immediately follow the write to ADCSC1 with a wait instruction or stop instruction.
 - For stop3 mode operation, select ADACK as the clock source. Operation in stop3 reduces V_{DD} noise but increases effective conversion time due to stop recovery.
- There is no I/O switching, input or output, on the MCU during the conversion.

There are some situations where external system activity causes radiated or conducted noise emissions or excessive V_{DD} noise is coupled into the ADC. In these situations, or when the MCU cannot be placed in wait or stop3 or I/O activity cannot be halted, these recommended actions may reduce the effect of noise on the accuracy:

- Place a 0.01 μF capacitor (C_{AS}) on the selected input channel to V_{REFL} or V_{SSAD} (this improves noise issues, but affects the sample rate based on the external analog source resistance).
- Average the result by converting the analog input many times in succession and dividing the sum of the results. Four samples are required to eliminate the effect of a 1LSB, one-time error.
- Reduce the effect of synchronous noise by operating off the asynchronous clock (ADACK) and averaging. Noise that is synchronous to ADCK cannot be averaged out.

10.6.2.4 Code Width and Quantization Error

The ADC quantizes the ideal straight-line transfer function into 4096 steps (in 12-bit mode). Each step ideally has the same height (1 code) and width. The width is defined as the delta between the transition points to one code and the next. The ideal code width for an N bit converter (in this case N can be 8, 10 or 12), defined as 1LSB, is:

$$1 \text{ lsb} = (V_{\text{REFH}} - V_{\text{REFL}}) / 2^N \quad \text{Eqn. 10-2}$$

There is an inherent quantization error due to the digitization of the result. For 8-bit or 10-bit conversions the code transitions when the voltage is at the midpoint between the points where the straight line transfer function is exactly represented by the actual transfer function. Therefore, the quantization error will be $\pm 1/2$ lsb in 8- or 10-bit mode. As a consequence, however, the code width of the first (0x000) conversion is only 1/2 lsb and the code width of the last (0xFF or 0x3FF) is 1.5 lsb.

Table 11-8. IICC2 Field Descriptions

Field	Description
7 GCAEN	General Call Address Enable. The GCAEN bit enables or disables general call address. 0 General call address is disabled 1 General call address is enabled
6 ADEXT	Address Extension. The ADEXT bit controls the number of bits used for the slave address. 0 7-bit address scheme 1 10-bit address scheme
2–0 AD[10:8]	Slave Address. The AD[10:8] field contains the upper three bits of the slave address in the 10-bit address scheme. This field is only valid when the ADEXT bit is set.

11.4 Functional Description

This section provides a complete functional description of the IIC module.

11.4.1 IIC Protocol

The IIC bus system uses a serial data line (SDA) and a serial clock line (SCL) for data transfer. All devices connected to it must have open drain or open collector outputs. A logic AND function is exercised on both lines with external pull-up resistors. The value of these resistors is system dependent.

Normally, a standard communication is composed of four parts:

- Start signal
- Slave address transmission
- Data transfer
- Stop signal

The stop signal should not be confused with the CPU stop instruction. The IIC bus system communication is described briefly in the following sections and illustrated in [Figure 11-9](#).

Table 12-2. CANCTL1 Register Field Descriptions (continued)

Field	Description
1 SLPAK	Sleep Mode Acknowledge — This flag indicates whether the MSCAN module has entered sleep mode (see Section 12.5.5.4, “MSCAN Sleep Mode”). It is used as a handshake flag for the SLPRQ sleep mode request. Sleep mode is active when SLPRQ = 1 and SLPAK = 1. Depending on the setting of WUPE, the MSCAN will clear the flag if it detects activity on the CAN bus while in sleep mode. CPU clearing the SLPRQ bit will also reset the SLPAK bit. 0 Running — The MSCAN operates normally 1 Sleep mode active — The MSCAN has entered sleep mode
0 INITAK	Initialization Mode Acknowledge — This flag indicates whether the MSCAN module is in initialization mode (see Section 12.5.5.5, “MSCAN Initialization Mode”). It is used as a handshake flag for the INITRQ initialization mode request. Initialization mode is active when INITRQ = 1 and INITAK = 1. The registers CANCTL1, CANBTR0, CANBTR1, CANIDAC, CANIDAR0–CANIDAR7, and CANIDMR0–CANIDMR7 can be written only by the CPU when the MSCAN is in initialization mode. 0 Running — The MSCAN operates normally 1 Initialization mode active — The MSCAN is in initialization mode

12.3.3 MSCAN Bus Timing Register 0 (CANBTR0)

The CANBTR0 register configures various CAN bus timing parameters of the MSCAN module.

	7	6	5	4	3	2	1	0
R								
W								
	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
Reset:	0	0	0	0	0	0	0	0

Figure 12-6. MSCAN Bus Timing Register 0 (CANBTR0)

Read: Anytime

Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1)

Table 12-3. CANBTR0 Register Field Descriptions

Field	Description
7:6 SJW[1:0]	Synchronization Jump Width — The synchronization jump width defines the maximum number of time quanta (Tq) clock cycles a bit can be shortened or lengthened to achieve resynchronization to data transitions on the CAN bus (see Table 12-4).
5:0 BRP[5:0]	Baud Rate Prescaler — These bits determine the time quanta (Tq) clock which is used to build up the bit timing (see Table 12-5).

Table 12-4. Synchronization Jump Width

SJW1	SJW0	Synchronization Jump Width
0	0	1 Tq clock cycle
0	1	2 Tq clock cycles
1	0	3 Tq clock cycles
1	1	4 Tq clock cycles

Table 12-20. CANIDAR0–CANIDAR3 Register Field Descriptions

Field	Description
7:0 AC[7:0]	Acceptance Code Bits — AC[7:0] comprise a user-defined sequence of bits with which the corresponding bits of the related identifier register (IDRn) of the receive message buffer are compared. The result of this comparison is then masked with the corresponding identifier mask register.

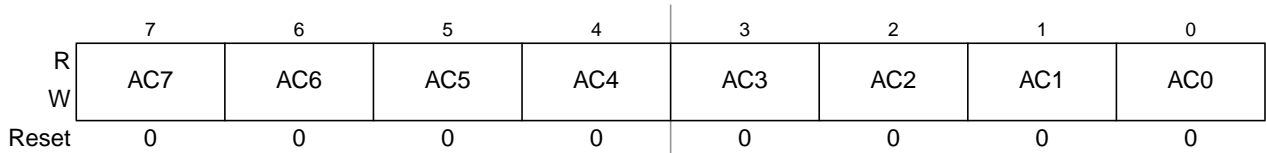


Figure 12-20. MSCAN Identifier Acceptance Registers (Second Bank) — CANIDAR4–CANIDAR7

Read: Anytime

Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1)

Table 12-21. CANIDAR4–CANIDAR7 Register Field Descriptions

Field	Description
7:0 AC[7:0]	Acceptance Code Bits — AC[7:0] comprise a user-defined sequence of bits with which the corresponding bits of the related identifier register (IDRn) of the receive message buffer are compared. The result of this comparison is then masked with the corresponding identifier mask register.

12.3.16 MSCAN Identifier Mask Registers (CANIDMR0–CANIDMR7)

The identifier mask register specifies which of the corresponding bits in the identifier acceptance register are relevant for acceptance filtering. To receive standard identifiers in 32 bit filter mode, it is required to program the last three bits (AM[2:0]) in the mask registers CANIDMR1 and CANIDMR5 to “don’t care.” To receive standard identifiers in 16 bit filter mode, it is required to program the last three bits (AM[2:0]) in the mask registers CANIDMR1, CANIDMR3, CANIDMR5, and CANIDMR7 to “don’t care.”

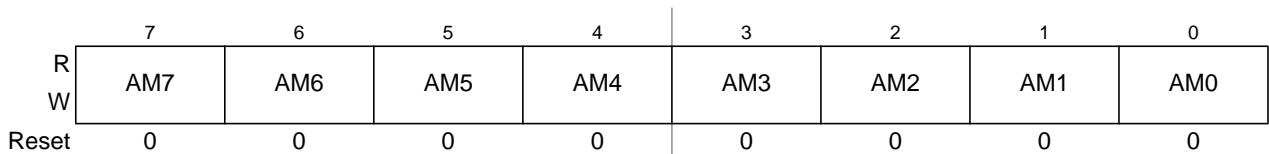


Figure 12-21. MSCAN Identifier Mask Registers (First Bank) — CANIDMR0–CANIDMR3

Read: Anytime

Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1)

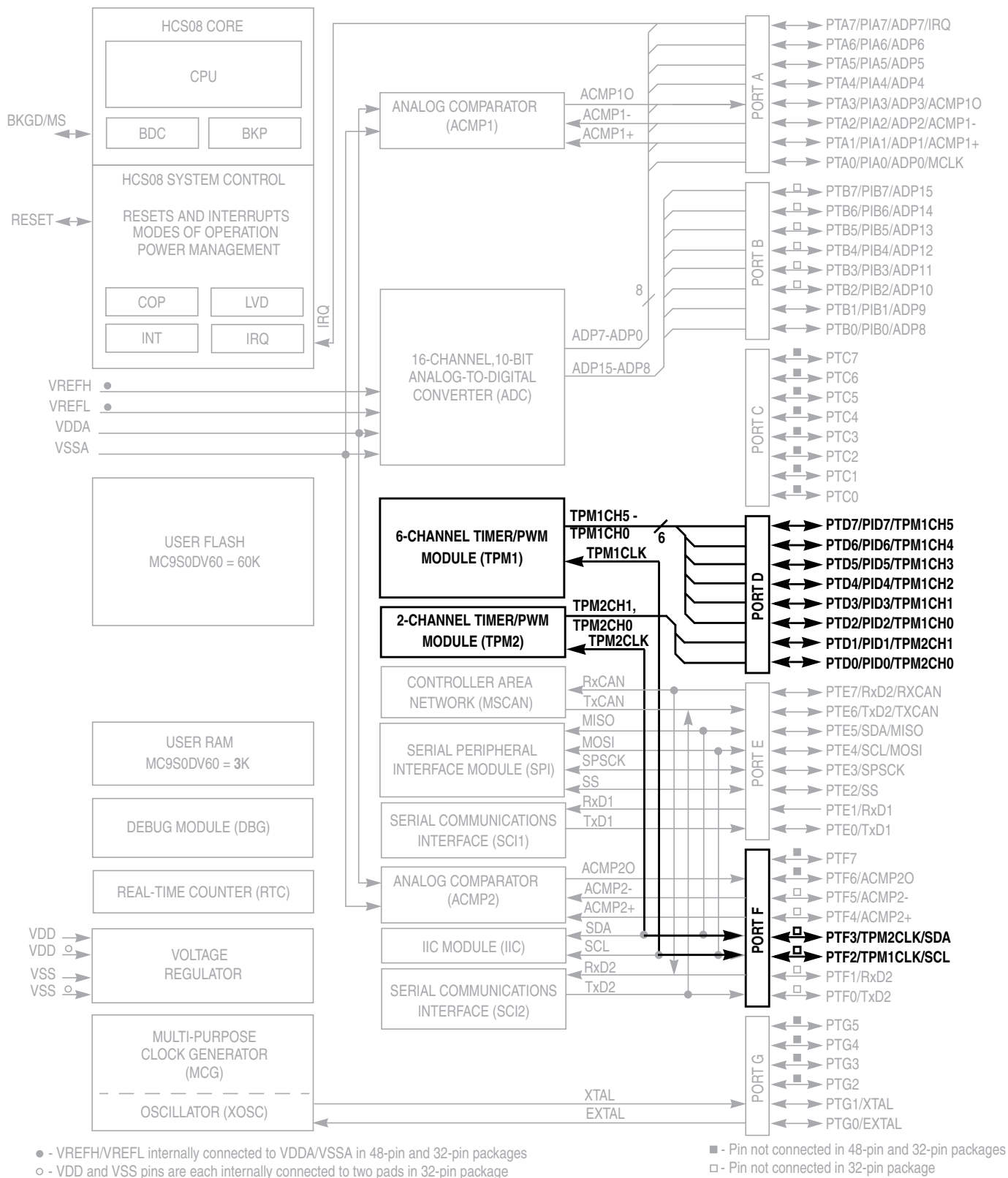


Figure 16-1. MC9S08DV60 Block Diagram

When a channel is configured for edge-aligned PWM (CPWMS=0, MSnB=1 and ELSnB:ELSnA not = 0:0), the data direction is overridden, the TPMxCHn pin is forced to be an output controlled by the TPM, and ELSnA controls the polarity of the PWM output signal on the pin. When ELSnB:ELSnA=1:0, the TPMxCHn pin is forced high at the start of each new period (TPMxCNT=0x0000), and the pin is forced low when the channel value register matches the timer counter. When ELSnA=1, the TPMxCHn pin is forced low at the start of each new period (TPMxCNT=0x0000), and the pin is forced high when the channel value register matches the timer counter.

TPMxMODH:TPMxMODL = 0x0008
 TPMxCnVH:TPMxCnVL = 0x0005

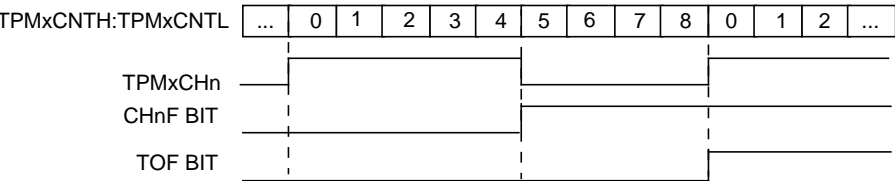


Figure 16-3. High-True Pulse of an Edge-Aligned PWM

TPMxMODH:TPMxMODL = 0x0008
 TPMxCnVH:TPMxCnVL = 0x0005

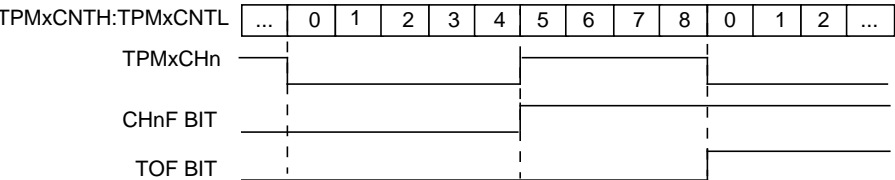


Figure 16-4. Low-True Pulse of an Edge-Aligned PWM

16.6.2.1.2 Center-Aligned PWM Case

When CPWMS=1, TOF gets set when the timer counter changes direction from up-counting to down-counting at the end of the terminal count (the value in the modulo register). In this case the TOF corresponds to the end of a PWM period.

16.6.2.2 Channel Event Interrupt Description

The meaning of channel interrupts depends on the channel's current mode (input-capture, output-compare, edge-aligned PWM, or center-aligned PWM).

16.6.2.2.1 Input Capture Events

When a channel is configured as an input capture channel, the ELSnB:ELSnA control bits select no edge (off), rising edges, falling edges or any edge as the edge which triggers an input capture event. When the selected edge is detected, the interrupt flag is set. The flag is cleared by the two-step sequence described in [Section 16.6.2, "Description of Interrupt Operation."](#)

16.6.2.2.2 Output Compare Events

When a channel is configured as an output compare channel, the interrupt flag is set each time the main timer counter matches the 16-bit value in the channel value register. The flag is cleared by the two-step sequence described [Section 16.6.2, "Description of Interrupt Operation."](#)

16.6.2.2.3 PWM End-of-Duty-Cycle Events

For channels configured for PWM operation there are two possibilities. When the channel is configured for edge-aligned PWM, the channel flag gets set when the timer counter matches the channel value register which marks the end of the active duty cycle period. When the channel is configured for center-aligned PWM, the timer count matches the channel value register twice during each PWM cycle. In this CPWM case, the channel flag is set at the start and at the end of the active duty cycle period which are the times when the timer counter matches the channel value register. The flag is cleared by the two-step sequence described [Section 16.6.2, "Description of Interrupt Operation."](#)

16.7 The Differences from TPM v2 to TPM v3

1. Write to TPMxCNTH:L registers ([Section 16.3.2, "TPM-Counter Registers \(TPMxCNTH:TPMxCNTL\)"](#)) [SE110-TPM case 7]
Any write to TPMxCNTH or TPMxCNTL registers in TPM v3 clears the TPM counter (TPMxCNTH:L) and the prescaler counter. Instead, in the TPM v2 only the TPM counter is cleared in this case.
2. Read of TPMxCNTH:L registers ([Section 16.3.2, "TPM-Counter Registers \(TPMxCNTH:TPMxCNTL\)"](#))
 - In TPM v3, any read of TPMxCNTH:L registers during BDM mode returns the value of the TPM counter that is frozen. In TPM v2, if only one byte of the TPMxCNTH:L registers was read before the BDM mode became active, then any read of TPMxCNTH:L registers during

BDM mode returns the latched value of TPMxCNTH:L from the read buffer instead of the frozen TPM counter value.

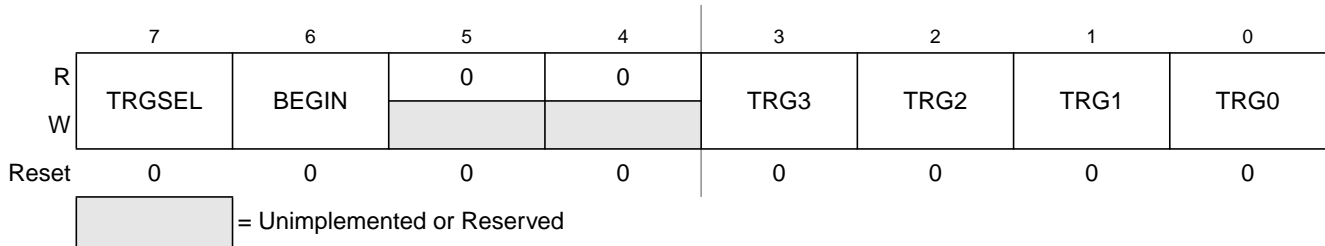
- This read coherency mechanism is cleared in TPM v3 in BDM mode if there is a write to TPMxSC, TPMxCNTH or TPMxCNTL. Instead, in these conditions the TPM v2 does not clear this read coherency mechanism.
3. Read of TPMxCnVH:L registers (Section 16.3.5, “TPM Channel Value Registers (TPMxCnVH:TPMxCnVL))
- In TPM v3, any read of TPMxCnVH:L registers during BDM mode returns the value of the TPMxCnVH:L register. In TPM v2, if only one byte of the TPMxCnVH:L registers was read before the BDM mode became active, then any read of TPMxCnVH:L registers during BDM mode returns the latched value of TPMxCNTH:L from the read buffer instead of the value in the TPMxCnVH:L registers.
 - This read coherency mechanism is cleared in TPM v3 in BDM mode if there is a write to TPMxCnSC. Instead, in this condition the TPM v2 does not clear this read coherency mechanism.
4. Write to TPMxCnVH:L registers
- Input Capture Mode (Section 16.4.2.1, “Input Capture Mode)
- In this mode the TPM v3 does not allow the writes to TPMxCnVH:L registers. Instead, the TPM v2 allows these writes.
- Output Compare Mode (Section 16.4.2.2, “Output Compare Mode)
- In this mode and if (CLKSB:CLKSA not = 0:0), the TPM v3 updates the TPMxCnVH:L registers with the value of their write buffer at the next change of the TPM counter (end of the prescaler counting) after the second byte is written. Instead, the TPM v2 always updates these registers when their second byte is written.
- The following procedure can be used in the TPM v3 to verify if the TPMxCnVH:L registers were updated with the new value that was written to these registers (value in their write buffer).
- ```

...
write the new value to TPMxCnVH:L;
read TPMxCnVH and TPMxCnVL registers;
while (the read value of TPMxCnVH:L is different from the new value written to
TPMxCnVH:L)
begin
 read again TPMxCnVH and TPMxCnVL;
end
...

```
- In this point, the TPMxCnVH:L registers were updated, so the program can continue and, for example, write to TPMxC0SC without cancelling the previous write to TPMxCnVH:L registers.
- Edge-Aligned PWM (Section 16.4.2.3, “Edge-Aligned PWM Mode)
- In this mode and if (CLKSB:CLKSA not = 00), the TPM v3 updates the TPMxCnVH:L registers with the value of their write buffer after that the both bytes were written and when the

### 17.4.3.8 Debug Trigger Register (DBGT)

This register can be read any time, but may be written only if ARM = 0, except bits 4 and 5 are hard-wired to 0s.



**Figure 17-8. Debug Trigger Register (DBGT)**

**Table 17-5. DBGT Register Field Descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7<br>TRGSEL     | <b>Trigger Type</b> — Controls whether the match outputs from comparators A and B are qualified with the opcode tracking logic in the debug module. If TRGSEL is set, a match signal from comparator A or B must propagate through the opcode tracking logic and a trigger event is only signalled to the FIFO logic if the opcode at the match address is actually executed.<br>0 Trigger on access to compare address (force)<br>1 Trigger if opcode at compare address is executed (tag) |
| 6<br>BEGIN      | <b>Begin/End Trigger Select</b> — Controls whether the FIFO starts filling at a trigger or fills in a circular manner until a trigger ends the capture of information. In event-only trigger modes, this bit is ignored and all debug runs are assumed to be begin traces.<br>0 Data stored in FIFO until trigger (end trace)<br>1 Trigger initiates data storage (begin trace)                                                                                                             |
| 3:0<br>TRG[3:0] | <b>Select Trigger Mode</b> — Selects one of nine triggering modes, as described below.<br>0000 A-only<br>0001 A OR B<br>0010 A Then B<br>0011 Event-only B (store data)<br>0100 A then event-only B (store data)<br>0101 A AND B data (full mode)<br>0110 A AND NOT B data (full mode)<br>0111 Inside range: $A \leq \text{address} \leq B$<br>1000 Outside range: $\text{address} < A$ or $\text{address} > B$<br>1001 – 1111 (No trigger)                                                 |

at the transition from the value set in the modulus register and the next lower count value. This corresponds to the end of a PWM period. (The 0x0000 count value corresponds to the center of a period.)

## B.4.3 Channel Event Interrupt Description

The meaning of channel interrupts depends on the current mode of the channel (input capture, output compare, edge-aligned PWM, or center-aligned PWM).

When a channel is configured as an input capture channel, the ELSnB:ELSnA control bits select rising edges, falling edges, any edge, or no edge (off) as the edge that triggers an input capture event. When the selected edge is detected, the interrupt flag is set. The flag is cleared by the 2-step sequence described in [Section B.4.1, “Clearing Timer Interrupt Flags.”](#)

When a channel is configured as an output compare channel, the interrupt flag is set each time the main timer counter matches the 16-bit value in the channel value register. The flag is cleared by the 2-step sequence described in [Section B.4.1, “Clearing Timer Interrupt Flags.”](#)

## B.4.4 PWM End-of-Duty-Cycle Events

For channels that are configured for PWM operation, there are two possibilities:

- When the channel is configured for edge-aligned PWM, the channel flag is set when the timer counter matches the channel value register that marks the end of the active duty cycle period.
- When the channel is configured for center-aligned PWM, the timer count matches the channel value register twice during each PWM cycle. In this CPWM case, the channel flag is set at the start and at the end of the active duty cycle, which are the times when the timer counter matches the channel value register.

The flag is cleared by the 2-step sequence described in [Section B.4.1, “Clearing Timer Interrupt Flags.”](#)