

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	Active
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	CANbus, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	37
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68908gz8mfae

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0		
\$0030	Timer 2 Channel 0 Status and Control Register (T2SC0) See page 267.	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX	
		Write:	0								
		Reset:	0	0	0	0	0	0	0	0	
\$0031	Timer 2 Channel 0 Register High (T2CH0H) See page 270.	Read:	Bit 15	14	13	12	11	10	9	Bit 8	
		Write:									
		Reset:	Indeterminate after reset								
\$0032	Timer 2 Channel 0 Register Low (T2CH0L) See page 270.	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									
		Reset:	Indeterminate after reset								
\$0033	Timer 2 Channel 1 Status and Control Register (T2SC1) See page 267.	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX	
		Write:	0								
		Reset:	0	0	0	0	0	0	0	0	
\$0034	Timer 2 Channel 1 Register High (T2CH1H) See page 270.	Read:	Bit 15	14	13	12	11	10	9	Bit 8	
		Write:									
		Reset:	Indeterminate after reset								
\$0035	Timer 2 Channel 1 Register Low (T2CH1L) See page 270.	Read:	Bit 7	6	5	4	3	2	1	Bit 0	
		Write:									
		Reset:	Indeterminate after reset								
\$0036	PLL Control Register (PCTL) See page 69.	Read:	PLLIE	PLLF	PLLON	BCS	R	R	VPR1	VPR0	
		Write:									
		Reset:	0	0	1	0	0	0	0	0	
\$0037	PLL Bandwidth Control Register (PBWC) See page 71.	Read:	AUTO	LOCK	\overline{ACQ}	0	0	0	0	R	
		Write:									
		Reset:	0	0	0	0	0	0	0	0	
\$0038	PLL Multiplier Select High Register (PMSH) See page 72.	Read:	0	0	0	0	MUL11	MUL10	MUL9	MUL8	
		Write:									
		Reset:	0	0	0	0	0	0	0	0	
\$0039	PLL Multiplier Select Low Register (PMSL) See page 73.	Read:	MUL7	MUL6	MUL5	MUL4	MUL3	MUL2	MUL1	MUL0	
		Write:									
		Reset:	0	0	0	0	U	U	U	U	
\$003A	PLL VCO Select Range Register (PMRS) See page 73.	Read:	VRS7	VRS6	VRS5	VRS4	VRS3	VRS2	VRS1	VRS0	
		Write:									
		Reset:	0	1	0	0	0	0	0	0	
\$003B	Reserved	Read:	0	0	0	0	R	R	R	R	
		Write:									
		Reset:	0	0	0	0	0	0	0	1	

= Unimplemented R = Reserved U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 5 of 8)

Memory

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$003C	ADC Status and Control Register (ADSCR) See page 53.	Read:	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
		Write:								
		Reset:	0	0	0	1	1	1	1	1
\$003D	ADC Data High Register (ADRH) See page 55.	Read:	0	0	0	0	0	0	AD9	AD8
		Write:								
		Reset:	Unaffected by reset							
\$003E	ADC Data Low Register (ADRL) See page 55.	Read:	AD7	AD6	AD5	AD4	A3	AD2	AD1	AD0
		Write:								
		Reset:	Unaffected by reset							
\$003F	ADC Clock Register (ADCLK) See page 57.	Read:	ADIV2	ADIV1	ADIV0	ADICLK	MODE1	MODE0	R	0
		Write:								
		Reset:	0	0	0	0	0	1	0	0
\$0500 ↓ \$0508	MSCAN08 Control Registers See page 141.	MSCAN08 control registers (9 bytes) Refer to 12.13 Programmer's Model of Control Registers								
\$0509 ↓ \$050D	Reserved	Reserved (5 bytes)								
\$050E ↓ \$050F	MSCAN08 Error Counters	MSCAN08 error counters (2 bytes)								
\$0510 ↓ \$0517	MSCAN08 Identifier Filter See page 141.	MSCAN08 control registers (9 bytes) Refer to 12.13 Programmer's Model of Control Registers								
\$0518 ↓ \$053F	Reserved	Reserved (40 bytes)								
\$0540 ↓ \$054F	MSCAN08 Receive Buffer See page 137.	MSC08 receive buffer Refer to 12.12 Programmer's Model of Message Storage								
\$0550 ↓ \$055F	MSCAN08 Transmit Buffer 0 See page 137.	MSC08 transmitter buffer 0 Refer to 12.12 Programmer's Model of Message Storage								

= Unimplemented R = Reserved U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 6 of 8)

Memory

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$FE0A	Break Address Register Low (BRKL) See page 274.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FE0B	Break Status and Control Register (BRKSCR) See page 274.	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FE0C	LVI Status Register (LVISR) See page 119.	Read:	LVIOUT	0	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FF7E	FLASH Block Protect Register (FLBPR) ⁽³⁾ See page 44.	Read:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
		Write:								
		Reset:	Unaffected by reset							
3. Nonvolatile FLASH register										
\$FFFF	COP Control Register (COPCTL) See page 85.	Read:	Low byte of reset vector							
		Write:	Writing clears COP counter (any value)							
		Reset:	Unaffected by reset							

= Unimplemented R = Reserved U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 8 of 8)

The address ranges for the user memory and vectors are:

- \$C000–\$FDFF; user memory
- \$FE08; FLASH control register
- \$FF7E; FLASH block protect register
- \$FFD4–\$FFFF; these locations are reserved for user-defined interrupt and reset vectors

Programming tools are available from Freescale Semiconductor. Contact your local representative for more information.

NOTE

A security feature prevents viewing of the FLASH contents.⁽¹⁾

2.6.2 FLASH Control Register

The FLASH control register (FLCR) controls FLASH program and erase operations.

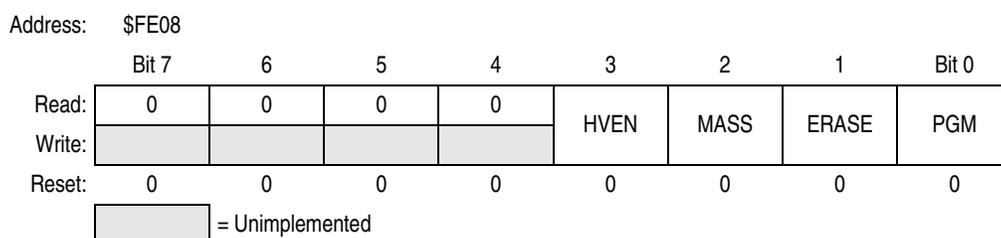


Figure 2-3. FLASH Control Register (FLCR)

HVEN — High-Voltage Enable Bit

This read/write bit enables the charge pump to drive high voltages for program and erase operations in the array. HVEN can only be set if either PGM = 1 or ERASE = 1 and the proper sequence for program or erase is followed.

- 1 = High voltage enabled to array and charge pump on
- 0 = High voltage disabled to array and charge pump off

MASS — Mass Erase Control Bit

Setting this read/write bit configures the 16-Kbyte FLASH array for mass erase operation.

- 1 = MASS erase operation selected
- 0 = PAGE erase operation selected

ERASE — Erase Control Bit

This read/write bit configures the memory for erase operation. ERASE is interlocked with the PGM bit such that both bits cannot be equal to 1 or set to 1 at the same time.

- 1 = Erase operation selected
- 0 = Erase operation unselected

PGM — Program Control Bit

This read/write bit configures the memory for program operation. PGM is interlocked with the ERASE bit such that both bits cannot be equal to 1 or set to 1 at the same time.

- 1 = Program operation selected
- 0 = Program operation unselected

1. No security feature is absolutely secure. However, Freescale’s strategy is to make reading or copying the FLASH difficult for unauthorized users.

The following conditions apply when in manual mode:

- \overline{ACQ} is a writable control bit that controls the mode of the filter. Before turning on the PLL in manual mode, the \overline{ACQ} bit must be clear.
- Before entering tracking mode ($\overline{ACQ} = 1$), software must wait a given time, t_{ACQ} (See 4.8 Acquisition/Lock Time Specifications.), after turning on the PLL by setting PLLON in the PLL control register (PCTL).
- Software must wait a given time, t_{AL} , after entering tracking mode before selecting the PLL as the clock source to CGMOUT (BCS = 1).
- The LOCK bit is disabled.
- CPU interrupts from the CGM are disabled.

4.3.6 Programming the PLL

Use the following procedure to program the PLL. For reference, the variables used and their meaning are shown in Table 4-1.

Table 4-1. Variable Definitions

Variable	Definition
f_{BUSDES}	Desired bus clock frequency
$f_{VCLKDES}$	Desired VCO clock frequency
f_{RCLK}	Chosen reference crystal frequency
f_{VCLK}	Calculated VCO clock frequency
f_{BUS}	Calculated bus clock frequency
f_{NOM}	Nominal VCO center frequency
f_{VRS}	Programmed VCO center frequency

NOTE

The round function in the following equations means that the real number should be rounded to the nearest integer number.

1. Choose the desired bus frequency, f_{BUSDES} .
2. Calculate the desired VCO frequency (four times the desired bus frequency).

$$f_{VCLKDES} = 4 \times f_{BUSDES}$$

3. Choose a practical PLL (crystal) reference frequency, f_{RCLK} . Typically, the reference crystal is 1–8 MHz.

Frequency errors to the PLL are corrected at a rate of f_{RCLK} .

For stability and lock time reduction, this rate must be as fast as possible. The VCO frequency must be an integer multiple of this rate. The relationship between the VCO frequency, f_{VCLK} , and the reference frequency, f_{RCLK} , is:

$$f_{VCLK} = (N) (f_{RCLK})$$

N, the range multiplier, must be an integer.

4.3.9 CGM External Connections

In its typical configuration, the CGM requires external components. Five of these are for the crystal oscillator and two or four are for the PLL.

The crystal oscillator is normally connected in a Pierce oscillator configuration, as shown in Figure 4-2. Figure 4-2 shows only the logical representation of the internal components and may not represent actual circuitry. The oscillator configuration uses five components:

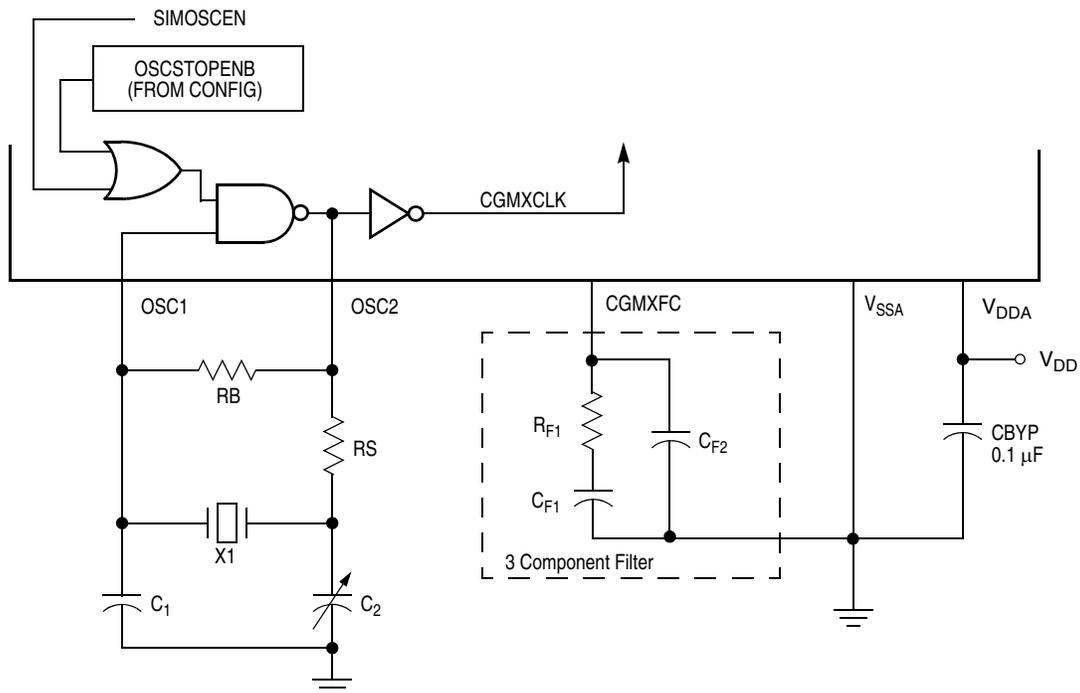
- Crystal, X_1
- Fixed capacitor, C_1
- Tuning capacitor, C_2 (can also be a fixed capacitor)
- Feedback resistor, R_B
- Series resistor, R_S

The series resistor (R_S) is included in the diagram to follow strict Pierce oscillator guidelines. Refer to the crystal manufacturer's data for more information regarding values for C_1 and C_2 .

Figure 4-2 also shows the external components for the PLL:

- Bypass capacitor, C_{BYP}
- Filter network

Routing should be done with great care to minimize signal cross talk and noise.



Note: Filter network in box can be replaced with a single capacitor, but will degrade stability.

Figure 4-2. CGM External Connections

4.5.4 PLL Multiplier Select Register Low

The PLL multiplier select register low (PMSL) contains the programming information for the low byte of the modulo feedback divider.

Address:	\$0038							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	MUL7	MUL6	MUL5	MUL4	MUL3	MUL2	MUL1	MUL0
Write:								
Reset:	0	1	0	0	0	0	0	0

Figure 4-7. PLL Multiplier Select Register Low (PMSL)

NOTE

For applications using 1–8 MHz reference frequencies this register must be reprogrammed before enabling the PLL. The reset value of this register will cause applications using 1–8 MHz reference frequencies to become unstable if the PLL is enabled without programming an appropriate value. The programmed value must not allow the VCO clock to exceed 32 MHz. See 4.3.6 Programming the PLL for detailed instructions on choosing the proper value for PMSL.

MUL7–MUL0 — Multiplier Select Bits

These read/write bits control the low byte of the modulo feedback divider that selects the VCO frequency multiplier, N. (See 4.3.3 PLL Circuits and 4.3.6 Programming the PLL.) MUL7–MUL0 cannot be written when the PLLON bit in the PCTL is set. A value of \$0000 in the multiplier select registers configures the modulo feedback divider the same as a value of \$0001. Reset initializes the register to \$40 for a default multiply value of 64.

NOTE

The multiplier select bits have built-in protection such that they cannot be written when the PLL is on (PLLON = 1).

4.5.5 PLL VCO Range Select Register

NOTE

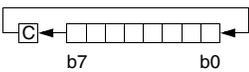
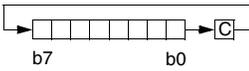
PMRS may be called PVRS on other HC08 derivatives.

The PLL VCO range select register (PMRS) contains the programming information required for the hardware configuration of the VCO.

Address:	\$003A							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	VRS7	VRS6	VRS5	VRS4	VRS3	VRS2	VRS1	VRS0
Write:								
Reset:	0	1	0	0	0	0	0	0

Figure 4-8. PLL VCO Range Select Register (PMRS)

Table 7-1. Instruction Set Summary (Sheet 5 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles	
			V	H	I	N	Z					C
PULA	Pull A from Stack	$SP \leftarrow (SP + 1); \text{Pull (A)}$	-	-	-	-	-	-	INH	86		2
PULH	Pull H from Stack	$SP \leftarrow (SP + 1); \text{Pull (H)}$	-	-	-	-	-	-	INH	8A		2
PULX	Pull X from Stack	$SP \leftarrow (SP + 1); \text{Pull (X)}$	-	-	-	-	-	-	INH	88		2
ROL <i>opr</i> ROLA ROLX ROL <i>opr,X</i> ROL ,X ROL <i>opr,SP</i>	Rotate Left through Carry		↑	-	-	↑	↑	↑	DIR INH INH IX1 IX SP1	39 49 59 69 79 9E69	dd ff ff	4 1 1 4 3 5
ROR <i>opr</i> RORA RORX ROR <i>opr,X</i> ROR ,X ROR <i>opr,SP</i>	Rotate Right through Carry		↑	-	-	↑	↑	↑	DIR INH INH IX1 IX SP1	36 46 56 66 76 9E66	dd ff ff	4 1 1 4 3 5
RSP	Reset Stack Pointer	$SP \leftarrow \$FF$	-	-	-	-	-	-	INH	9C		1
RTI	Return from Interrupt	$SP \leftarrow (SP + 1); \text{Pull (CCR)}$ $SP \leftarrow (SP + 1); \text{Pull (A)}$ $SP \leftarrow (SP + 1); \text{Pull (X)}$ $SP \leftarrow (SP + 1); \text{Pull (PCH)}$ $SP \leftarrow (SP + 1); \text{Pull (PCL)}$	↑	↑	↑	↑	↑	↑	INH	80		7
RTS	Return from Subroutine	$SP \leftarrow SP + 1; \text{Pull (PCH)}$ $SP \leftarrow SP + 1; \text{Pull (PCL)}$	-	-	-	-	-	-	INH	81		4
SBC # <i>opr</i> SBC <i>opr</i> SBC <i>opr</i> SBC <i>opr,X</i> SBC <i>opr,X</i> SBC ,X SBC <i>opr,SP</i> SBC <i>opr,SP</i>	Subtract with Carry	$A \leftarrow (A) - (M) - (C)$	↑	-	-	↑	↑	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	A2 B2 C2 D2 E2 F2 9EE2 9ED2	ii dd hh ll ee ff ff ff ff ff	2 3 4 4 3 2 4 5
SEC	Set Carry Bit	$C \leftarrow 1$	-	-	-	-	-	1	INH	99		1
SEI	Set Interrupt Mask	$I \leftarrow 1$	-	-	1	-	-	-	INH	9B		2
STA <i>opr</i> STA <i>opr</i> STA <i>opr,X</i> STA <i>opr,X</i> STA ,X STA <i>opr,SP</i> STA <i>opr,SP</i>	Store A in M	$M \leftarrow (A)$	0	-	-	↑	↑	-	DIR EXT IX2 IX1 IX SP1 SP2	B7 C7 D7 E7 F7 9EE7 9ED7	dd hh ll ee ff ff ff ff ff	3 4 4 3 2 4 5
STHX <i>opr</i>	Store H:X in M	$(M:M + 1) \leftarrow (H:X)$	0	-	-	↑	↑	-	DIR	35	dd	4
STOP	Enable Interrupts, Stop Processing, Refer to MCU Documentation	$I \leftarrow 0; \text{Stop Processing}$	-	-	0	-	-	-	INH	8E		1
STX <i>opr</i> STX <i>opr</i> STX <i>opr,X</i> STX <i>opr,X</i> STX ,X STX <i>opr,SP</i> STX <i>opr,SP</i>	Store X in M	$M \leftarrow (X)$	0	-	-	↑	↑	-	DIR EXT IX2 IX1 IX SP1 SP2	BF CF DF EF FF 9EEF 9EDF	dd hh ll ee ff ff ff ff ff	3 4 4 3 2 4 5
SUB # <i>opr</i> SUB <i>opr</i> SUB <i>opr</i> SUB <i>opr,X</i> SUB <i>opr,X</i> SUB ,X SUB <i>opr,SP</i> SUB <i>opr,SP</i>	Subtract	$A \leftarrow (A) - (M)$	↑	-	-	↑	↑	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	A0 B0 C0 D0 E0 F0 9EE0 9ED0	ii dd hh ll ee ff ff ff ff ff	2 3 4 4 3 2 4 5

MSCAN08 Controller (MSCAN08)

A double buffer scheme would de-couple the re-loading of the transmit buffers from the actual message being sent and as such reduces the reactivity requirements on the CPU. Problems may arise if the sending of a message would be finished just while the CPU re-loads the second buffer. In that case, no buffer would then be ready for transmission and the bus would be released.

At least three transmit buffers are required to meet the first of the above requirements under all circumstances. The MSCAN08 has three transmit buffers.

The second requirement calls for some sort of internal prioritization which the MSCAN08 implements with the "local priority" concept described in 12.4.2 Receive Structures.

12.4.2 Receive Structures

The received messages are stored in a 2-stage input first in first out (FIFO). The two message buffers are mapped using a "ping pong" arrangement into a single memory area (see Figure 12-3). While the background receive buffer (RxBG) is exclusively associated to the MSCAN08, the foreground receive buffer (RxFG) is addressable by the central processor unit (CPU08). This scheme simplifies the handler software, because only one address area is applicable for the receive process.

Both buffers have a size of 13 bytes to store the CAN control bits, the identifier (standard or extended), and the data content. For details, see 12.12 Programmer's Model of Message Storage.

The receiver full flag (RXF) in the MSCAN08 receiver flag register (CRFLG), signals the status of the foreground receive buffer. When the buffer contains a correctly received message with matching identifier, this flag is set. See 12.13.5 MSCAN08 Receiver Flag Register (CRFLG)

On reception, each message is checked to see if it passes the filter (for details see 12.5 Identifier Acceptance Filter) and in parallel is written into RxBG. The MSCAN08 copies the content of RxBG into RxFG⁽¹⁾, sets the RXF flag, and generates a receive interrupt to the CPU⁽²⁾. The user's receive handler has to read the received message from RxFG and to reset the RXF flag to acknowledge the interrupt and to release the foreground buffer. A new message which can follow immediately after the IFS field of the CAN frame, is received into RxBG. The overwriting of the background buffer is independent of the identifier filter function.

When the MSCAN08 module is transmitting, the MSCAN08 receives its own messages into the background receive buffer, RxBG. It does NOT overwrite RxFG, generate a receive interrupt or acknowledge its own messages on the CAN bus. The exception to this rule is in loop-back mode (see 12.13.2 MSCAN08 Module Control Register 1), where the MSCAN08 treats its own messages exactly like all other incoming messages. The MSCAN08 receives its own transmitted messages in the event that it loses arbitration. If arbitration is lost, the MSCAN08 must be prepared to become the receiver.

An overrun condition occurs when both the foreground and the background receive message buffers are filled with correctly received messages with accepted identifiers and another message is correctly received from the bus with an accepted identifier. The latter message will be discarded and an error interrupt with overrun indication will be generated if enabled. The MSCAN08 is still able to transmit messages with both receive message buffers filled, but all incoming messages are discarded.

1. Only if the RXF flag is not set.

2. The receive interrupt will occur only if not masked. A polling scheme can be applied on RXF also.

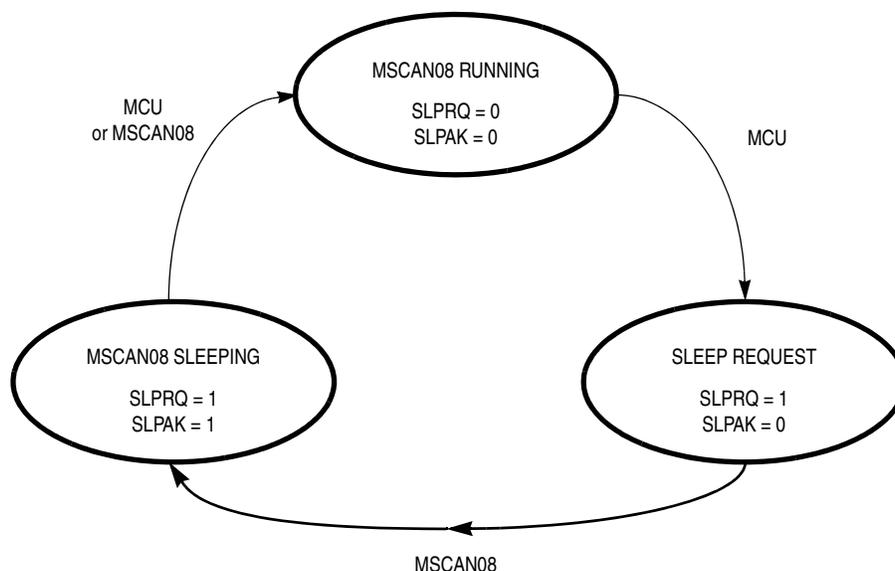


Figure 12-7. Sleep Request/Acknowledge Cycle

After wakeup, the MSCAN08 waits for 11 consecutive recessive bits to synchronize to the bus. As a consequence, if the MSCAN08 is woken-up by a CAN frame, this frame is not received. The receive message buffers (RxFG and RxBG) contain messages if they were received before sleep mode was entered. All pending actions are executed upon wakeup: copying of RxBG into RxFG, message aborts and message transmissions. If the MSCAN08 is still in bus-off state after sleep mode was left, it continues counting the 128*11 consecutive recessive bits.

12.8.2 MSCAN08 Soft Reset Mode

In soft reset mode, the MSCAN08 is stopped. Registers can still be accessed. This mode is used to initialize the module configuration, bit timing and the CAN message filter. See 12.13.1 MSCAN08 Module Control Register 0 for a complete description of the soft reset mode.

When setting the SFTRES bit, the MSCAN08 immediately stops all ongoing transmissions and receptions, potentially causing CAN protocol violations.

NOTE

The user is responsible to take care that the MSCAN08 is not active when soft reset mode is entered. The recommended procedure is to bring the MSCAN08 into sleep mode before the SFTRES bit is set.

12.8.3 MSCAN08 Power-Down Mode

The MSCAN08 is in power-down mode when the CPU is in stop mode.

When entering the power-down mode, the MSCAN08 immediately stops all ongoing transmissions and receptions, potentially causing CAN protocol violations.

NOTE

The user is responsible to take care that the MSCAN08 is not active when power-down mode is entered. The recommended procedure is to bring the MSCAN08 into sleep mode before the STOP instruction is executed.

12.13 Programmer's Model of Control Registers

The programmer's model has been laid out for maximum simplicity and efficiency. Figure 12-15 gives an overview on the control register block of the MSCAN08.

Addr.	Register		Bit 7	6	5	4	3	2	1	Bit 0
\$0500	CMCR0	Read:	0	0	0	SYNCH	TLNKEN	SLPAK	SLPRQ	SFTRES
		Write:								
\$0501	CMCR1	Read:	0	0	0	0	0	LOOPB	WUPM	CLKSRC
		Write:								
\$0502	CBTR0	Read:								
		Write:	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
\$0503	CBTR1	Read:								
		Write:	SAMP	TSEG22	TSEG21	TSEG20	TSEG13	TSEG12	TSEG11	TSEG10
\$0504	CRFLG	Read:								
		Write:	WUPIF	RWRNIF	TWRNIF	RERRIF	TERRIF	BOFFIF	OVRIF	RXF
\$0505	CRIER	Read:								
		Write:	WUPIE	RWRNIE	TWRNIE	RERRIE	TERRIE	BOFFIE	OVRIE	RXFIE
\$0506	CTFLG	Read:	0	ABTAK2	ABTAK1	ABTAK0	0	TXE2	TXE1	TXE0
		Write:								
\$0507	CTCR	Read:	0	ABTRQ2	ABTRQ1	ABTRQ0	0	TXEIE2	TXEIE1	TXEIE0
		Write:								
\$0508	CIDAC	Read:	0	0	IDAM1	IDAM0	0	0	IDHIT1	IDHIT0
		Write:								
\$0509	Reserved	Read:								
		Write:	R	R	R	R	R	R	R	R
\$050E	CRXERR	Read:								
		Write:	RXERR7	RXERR6	RXERR5	RXERR4	RXERR3	RXERR2	RXERR1	RXERR0
\$050F	CTXERR	Read:								
		Write:	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
\$0510	CIDAR0	Read:								
		Write:	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
\$0511	CIDAR1	Read:								
		Write:	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0

= Unimplemented
 R = Reserved

Figure 12-15. MSCAN08 Control Register Structure

Figure 13-4 shows the port A I/O logic.

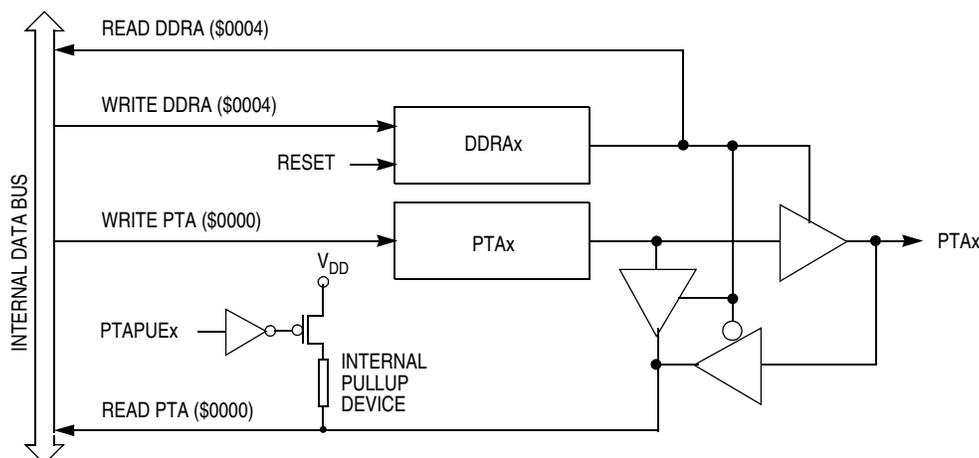


Figure 13-4. Port A I/O Circuit

When bit DDRA_x is a logic 1, reading address \$0000 reads the PTA_x data latch. When bit DDRA_x is a logic 0, reading address \$0000 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 13-2 summarizes the operation of the port A pins.

Table 13-2. Port A Pin Functions

PTAPUE Bit	DDRA Bit	PTA Bit	I/O Pin Mode	Accesses to DDRA			Accesses to PTA	
				Read/Write		Read	Write	
1	0	X ⁽¹⁾	Input, V _{DD} ⁽²⁾	DDRA7–DDRA0		Pin	PTA7–PTA0 ⁽³⁾	
0	0	X	Input, Hi-Z ⁽⁴⁾	DDRA7–DDRA0		Pin	PTA7–PTA0 ⁽³⁾	
X	1	X	Output	DDRA7–DDRA0		PTA7–PTA0	PTA7–PTA0	

1. X = Don't care
2. I/O pin pulled up to V_{DD} by internal pullup device
3. Writing affects data register, but does not affect input.
4. Hi-Z = High impedance

13.3.3 Port A Input Pullup Enable Register

The port A input pullup enable register (PTAPUE) contains a software configurable pullup device for each of the eight port A pins. Each bit is individually configurable and requires that the data direction register, DDRA, bit be configured as an input. Each pullup is automatically and dynamically disabled when a port bit's DDRA is configured for output mode.



Figure 13-5. Port A Input Pullup Enable Register (PTAPUE)

13.5.3 Port C Input Pullup Enable Register

The port C input pullup enable register (PTCPUE) contains a software configurable pullup device for each of the seven port C pins. Each bit is individually configurable and requires that the data direction register, DDRC, bit be configured as an input. Each pullup is automatically and dynamically disabled when a port bit's DDRC is configured for output mode.

Address: \$000E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	PTCPUE6	PTCPUE5	PTCPUE4	PTCPUE3	PTCPUE2	PTCPUE1	PTCPUE0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 13-12. Port C Input Pullup Enable Register (PTCPUE)

PTCPUE6–PTCPUE0 — Port C Input Pullup Enable Bits

These writable bits are software programmable to enable pullup devices on an input port bit.

- 1 = Corresponding port C pin configured to have internal pullup
- 0 = Corresponding port C pin internal pullup disconnected

13.6 Port D

Port D is an 8-bit special-function port that shares four of its pins with the serial peripheral interface (SPI) module and four of its pins with the two timer interface (TIM1 and TIM2) modules. Port D also has software configurable pullup devices if configured as an input port.

13.6.1 Port D Data Register

The port D data register (PTD) contains a data latch for each of the eight port D pins.

Address: \$0003

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
Write:								
Reset:	Unaffected by reset							
Alternative Function:	T2CH1	T2CH0	$\overline{T1CH1}$	T1CH0	SPSCK	\overline{MOSI}	MISO	\overline{SS}

Figure 13-13. Port D Data Register (PTD)

PTD7–PTD0 — Port D Data Bits

These read/write bits are software-programmable. Data direction of each port D pin is under the control of the corresponding bit in data direction register D. Reset has no effect on port D data.

T2CH1 and T2CH0 — Timer 2 Channel I/O Bits

The PTD7/T2CH1–PTD6/T2CH0 pins are the TIM2 input capture/output compare pins. The edge/level select bits, ELSxB:ELSxA, determine whether the PTD7/T2CH1–PTD6/T2CH0 pins are timer channel I/O pins or general-purpose I/O pins. See Chapter 19 Timer Interface Module (TIM).

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$0009	ESCI Prescaler Register (SCPSC) See page 206.	Read:	PDS2	PDS1	PDS0	PSSB4	PSSB3	PSSB2	PSSB1	PSSB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$000A	ESCI Arbiter Control Register (SCIACTL) See page 209.	Read:	AM1	Alost	AM0	ACLK	AFIN	ARUN	AROVFL	ARD8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$000B	ESCI Arbiter Data Register (SCIADAT) See page 210.	Read:	ARD7	ARD6	ARD5	ARD4	ARD3	ARD2	ARD1	ARD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0013	ESCI Control Register 1 (SCC1) See page 196.	Read:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0014	ESCI Control Register 2 (SCC2) See page 198.	Read:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0015	ESCI Control Register 3 (SCC3) See page 200.	Read:	R8	T8	R	R	ORIE	NEIE	FEIE	PEIE
		Write:								
		Reset:	U	0	0	0	0	0	0	0
\$0016	ESCI Status Register 1 (SCS1) See page 201.	Read:	SCTE	TC	SCRf	IDLE	OR	NF	FE	PE
		Write:								
		Reset:	1	1	0	0	0	0	0	0
\$0017	ESCI Status Register 2 (SCS2) See page 203.	Read:	0	0	0	0	0	0	BKF	RPF
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0018	ESCI Data Register (SCDR) See page 204.	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by reset							
\$0019	ESCI Baud Rate Register (SCBR) See page 204.	Read:	LINT	LINR	SCP1	SCP0	R	SCR2	SCR1	SCR0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented
R = Reserved
U = Unaffected

Figure 15-3. ESCI I/O Register Summary

The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit character with no errors is:

$$\left| \frac{170 - 163}{170} \right| \times 100 = 4.12\%$$

Fast Data Tolerance

Figure 15-9 shows how much a fast received character can be misaligned without causing a noise error or a framing error. The fast stop bit ends at RT10 instead of RT16 but is still there for the stop bit data samples at RT8, RT9, and RT10.

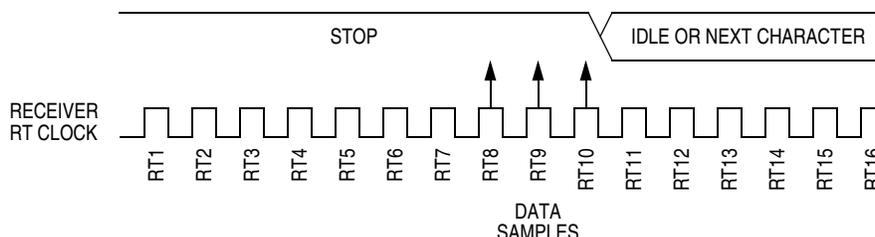


Figure 15-9. Fast Data

For an 8-bit character, data sampling of the stop bit takes the receiver $9 \text{ bit times} \times 16 \text{ RT cycles} + 10 \text{ RT cycles} = 154 \text{ RT cycles}$.

With the misaligned character shown in Figure 15-9, the receiver counts 154 RT cycles at the point when the count of the transmitting device is $10 \text{ bit times} \times 16 \text{ RT cycles} = 160 \text{ RT cycles}$.

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit character with no errors is

$$\left| \frac{154 - 160}{154} \right| \times 100 = 3.90\%$$

For a 9-bit character, data sampling of the stop bit takes the receiver $10 \text{ bit times} \times 16 \text{ RT cycles} + 10 \text{ RT cycles} = 170 \text{ RT cycles}$.

With the misaligned character shown in Figure 15-9, the receiver counts 170 RT cycles at the point when the count of the transmitting device is $11 \text{ bit times} \times 16 \text{ RT cycles} = 176 \text{ RT cycles}$.

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit character with no errors is:

$$\left| \frac{170 - 176}{170} \right| \times 100 = 3.53\%$$

15.4.3.6 Receiver Wakeup

So that the MCU can ignore transmissions intended only for other receivers in multiple-receiver systems, the receiver can be put into a standby state. Setting the receiver wakeup bit, RWU, in SCC2 puts the receiver into a standby state during which receiver interrupts are disabled.

Timer Interface Module (TIM)

If TIM functions are not required during wait mode, reduce power consumption by stopping the TIM before executing the WAIT instruction.

19.6.2 Stop Mode

The TIM is inactive after the execution of a STOP instruction. The STOP instruction does not affect register conditions or the state of the TIM counter. TIM operation resumes when the MCU exits stop mode after an external interrupt.

19.7 TIM During Break Interrupts

A break interrupt stops the TIM counter.

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state. See 16.7.3 Break Flag Control Register.

To allow software to clear status bits during a break interrupt, write a logic 1 to the BCFE bit. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect status bits during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), software can read and write I/O registers during the break state without affecting status bits. Some status bits have a 2-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is at logic 0. After the break, doing the second step clears the status bit.

19.8 I/O Signals

Port D shares four of its pins with the TIM. The four TIM channel I/O pins are T1CH0, T1CH1, T2CH0, and T2CH1 as described in 19.3 Pin Name Conventions.

Each channel I/O pin is programmable independently as an input capture pin or an output compare pin. T1CH0 and T2CH0 can be configured as buffered output compare or buffered PWM pins.

19.9 I/O Registers

NOTE

References to either timer 1 or timer 2 may be made in the following text by omitting the timer number. For example, TSC may generically refer to both T1SC AND T2SC.

These I/O registers control and monitor operation of the TIM:

- TIM status and control register (TSC)
- TIM counter registers (TCNTH:TCNTL)
- TIM counter modulo registers (TMODH:TMODL)
- TIM channel status and control registers (TSC0 and TSC1)
- TIM channel registers (TCH0H:TCH0L, TCH1H:TCH1L)

Chapter 20

Development Support

20.1 Introduction

This section describes the break module, the monitor read-only memory (MON), and the monitor mode entry methods.

20.2 Break Module (BRK)

The break module can generate a break interrupt that stops normal program flow at a defined address to enter a background program.

Features of the break module include:

- Accessible input/output (I/O) registers during the break Interrupt
- Central processor unit (CPU) generated break interrupts
- Software-generated break interrupts
- Computer operating properly (COP) disabling during break interrupts

20.2.1 Functional Description

When the internal address bus matches the value written in the break address registers, the break module issues a breakpoint signal ($\overline{\text{BKPT}}$) to the system integration module (SIM). The SIM then causes the CPU to load the instruction register with a software interrupt instruction (SWI). The program counter vectors to \$FFFC and \$FFFD (\$FEFC and \$FEFD in monitor mode).

The following events can cause a break interrupt to occur:

- A CPU generated address (the address in the program counter) matches the contents of the break address registers.
- Software writes a logic 1 to the BRKA bit in the break status and control register.

When a CPU generated address matches the contents of the break address registers, the break interrupt is generated. A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the microcontroller unit (MCU) to normal operation.

Figure 20-1 shows the structure of the break module.

Figure 20-2 provides a summary of the I/O registers.

Electrical Specifications

Characteristic ⁽¹⁾	Symbol	Min	Typ ⁽²⁾	Max	Unit
Monitor mode entry voltage	V_{TST}	$V_{DD} + 2.5$	—	$V_{DD} + 4.0$	V
Low-voltage inhibit, trip falling voltage	V_{TRIPF}	2.35	2.6	2.7	V
Low-voltage inhibit, trip rising voltage	V_{TRIPR}	2.4	2.66	2.8	V
Low-voltage inhibit reset/recover hysteresis ($V_{TRIPF} + V_{HYS} = V_{TRIPR}$)	V_{HYS}	—	100	—	mV
POR rearm voltage ⁽¹²⁾	V_{POR}	0	—	100	mV
POR reset voltage ⁽¹³⁾	V_{PORRST}	0	700	800	mV
POR rise time ramp rate ⁽¹⁴⁾	R_{POR}	0.035	—	—	V/ms

- $V_{DD} = 3.3 \text{ Vdc} \pm 10\%$, $V_{SS} = 0 \text{ Vdc}$, $T_A = T_A (\text{min})$ to $T_A (\text{max})$, unless otherwise noted
- Typical values reflect average measurements at midpoint of voltage range, 25°C only.
- Run (operating) I_{DD} measured using external square wave clock source ($f_{OSC} = 16 \text{ MHz}$). All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. $C_L = 20 \text{ pF}$ on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects run I_{DD} . Measured with all modules enabled.
- Wait I_{DD} measured using external square wave clock source ($f_{OSC} = 16 \text{ MHz}$). All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. $C_L = 20 \text{ pF}$ on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects wait I_{DD} . Measured with CGM and LVI enabled.
- Stop I_{DD} is measured with $OSC1 = V_{SS}$. All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. All ports configured as inputs. Typical values at midpoint of voltage range, 25°C only.
- Stop I_{DD} with TBM enabled is measured using an external square wave clock source ($f_{OSC} = 4 \text{ MHz}$). All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. All inputs configured as inputs.
- This parameter is characterized and not tested on each device.
- All functional non-supply pins are internally clamped to V_{SS} and V_{DD} .
- Input must be current limited to the value specified. To determine the value of the required current-limiting resistor, calculate resistance values for positive and negative clamp voltages, then use the larger of the two values.
- Power supply must maintain regulation within operating V_{DD} range during instantaneous and operating maximum current conditions. If positive injection current ($V_{in} > V_{DD}$) is greater than I_{DD} , the injection current may flow out of V_{DD} and could result in external power supply going out of regulation. Ensure external V_{DD} load will shunt current greater than maximum injection current. This will be the greatest risk when the MCU is not consuming power. Examples are: if no system clock is present, or if clock rate is very low (which would reduce overall power consumption).
- Pullups and pulldowns are disabled.
- Maximum is highest voltage that POR is guaranteed.
- Maximum is highest voltage that POR is possible.
- If minimum V_{DD} is not reached before the internal POR reset is released, \overline{RST} must be driven low externally until minimum V_{DD} is reached.

21.11 3.3-Volt ADC Characteristics

Characteristic ⁽¹⁾	Symbol	Min	Max	Unit	Comments
Supply voltage	V_{DDAD}	3.0	3.6	V	V_{DDAD} should be tied to the same potential as V_{DD} via separate traces.
Input voltages	V_{ADIN}	0	V_{DDAD}	V	$V_{ADIN} \leq V_{DDAD}$
Resolution	B_{AD}	10	10	Bits	
Absolute accuracy	A_{AD}	-6	+6	LSB	Includes quantization
ADC internal clock	f_{ADIC}	500 k	1.048 M	Hz	$t_{AIC} = 1/f_{ADIC}$
Conversion range	R_{AD}	V_{SSAD}	V_{DDAD}	V	
Power-up time	t_{ADPU}	16	—	t_{AIC} cycles	
Conversion time	t_{ADC}	16	17	t_{AIC} cycles	
Sample time	t_{ADS}	5	—	t_{AIC} cycles	
Monotonicity	M_{AD}	Guaranteed			
Zero input reading	Z_{ADI}	000	005	Hex	$V_{ADIN} = V_{SSA}$
Full-scale reading	F_{ADI}	3FA	3FF	Hex	$V_{ADIN} = V_{DDA}$
Input capacitance	C_{ADI}	—	30	pF	Not tested
V_{DDAD}/V_{REFH} current	I_{VREF}	—	1.2	mA	
Absolute accuracy (8-bit truncation mode)	A_{AD}	-1	+1	LSB	Includes quantization
Quantization error (8-bit truncation mode)	—	-1/8	+7/8	LSB	

1. $V_{DD} = 3.3 \text{ Vdc} \pm 10\%$, $V_{SS} = 0 \text{ Vdc}$, $V_{DDAD}/V_{REFH} = 3.3 \text{ Vdc} \pm 10\%$, $V_{SSAD}/V_{REFL} = 0 \text{ Vdc}$