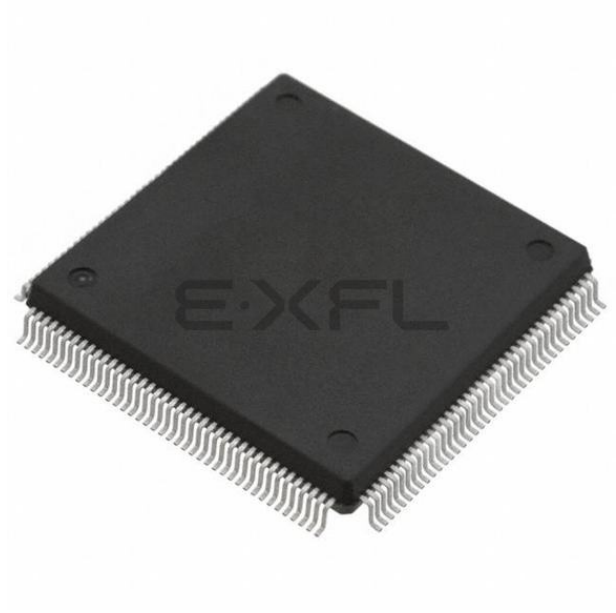

Welcome to E-XFL.COM

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	CPU32
Core Size	32-Bit Single-Core
Speed	20MHz
Connectivity	CANbus, EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	18
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	7.5K x 8
Voltage - Supply (Vcc/Vdd)	4.75V ~ 5.25V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	160-BQFP
Supplier Device Package	160-QFP (28x28)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/sc68376bgvab20

TABLE OF CONTENTS
(Continued)

Paragraph **Title** **Page**

LIST OF ILLUSTRATIONS
(Continued)

LIST OF TABLES
(Continued)

Table 3-5 MC68336/376 Signal Functions

Mnemonic	Signal Name	Function
ADDR[23:0]	Address Bus	24-bit address bus used by the CPU32
AN[59:48]/[3:0]	QADC Analog Input	16 channel A/D converter analog input pins
AN[w, x, y, z]	QADC Analog Input	Four input channels utilized when operating in multiplexed mode
$\overline{AS}$	Address Strobe	Indicates that a valid address is on the address bus
$\overline{AVEC}$	Autovector	Requests an automatic vector during interrupt acknowledge
$\overline{BERR}$	Bus Error	Indicates that a bus error has occurred
$\overline{BG}$	Bus Grant	Indicates that the MCU has relinquished the bus
$\overline{BGACK}$	Bus Grant Acknowledge	Indicates that an external device has assumed bus mastership
$\overline{BKPT}$	Breakpoint	Signals a hardware breakpoint to the CPU
$\overline{BR}$	Bus Request	Indicates that an external device requires bus mastership
CLKOUT	System Clock Out	System clock output
CANRX0	TouCAN Receive Data	CAN serial data input
CANTX0	TouCAN Transmit Data	CAN serial data output
$\overline{CS[10:0]}$	Chip-Selects	Select external devices at programmed addresses
$\overline{CSBOOT}$	Boot Chip-Select	Chip-select for external bootstrap memory
CPWM[8:5]	CTM4 PWMs	Four pulse-width modulation channels
CTD[10:9]/[4:3]	CTM4 Double Action Channels	Bidirectional double action timer channels
CTM2C	CTM4 Modulus Clock	Modulus counter clock input
DATA[15:0]	Data Bus	16-bit data bus used by the CPU32
$\overline{DS}$	Data Strobe	Indicates that an external device should place valid data on the data bus during a read cycle and that valid data has been placed on the data bus by the CPU during a write cycle.
$\overline{DSACK[1:0]}$	Data and Size Acknowledge	Provides asynchronous data transfers and dynamic bus sizing
DSI, DSO, DSCLK	Developmental Serial In, Out, Clock	Serial I/O and clock for background debug mode
ECLK	E-Clock	M6800 bus clock output
ETRIG[2:1]	QADC External Trigger	External trigger pins used when a QADC scan queue is in external trigger mode
EXTAL, XTAL	Crystal Oscillator	Connections for clock synthesizer circuit reference; a crystal or an external oscillator can be used
FC[2:0]	Function Codes	Identify processor state and current address space
FREEZE	Freeze	Indicates that the CPU has acknowledged a breakpoint
$\overline{HALT}$	Halt	Suspend external bus activity
$\overline{IFETCH}$	Instruction Pipeline	Indicates instruction pipeline activity
$\overline{IPIPE}$	Instruction Pipeline	Indicates instruction pipeline activity
$\overline{IRQ[7:1]}$	Interrupt Request	Requests an interrupt of specified priority level from the CPU
MA[2:0]	QADC Multiplexed Address	When external multiplexing is used, these pins provide addresses to the external multiplexer
MISO	Master In, Slave Out	Serial input to QSPI in the master mode; serial output from QSPI in the slave mode
MODCLK	Clock Mode Select	Selects the source of the system clock
MOSI	Master Out, Slave In	Serial output from the QSPI in master mode; serial input to the QSPI in slave mode
PC[6:0]	Port C	SIM digital output port signals
PCS[3:0]	Peripheral Chip-Selects	QSPI peripheral chip-select
PE[7:0]	Port E	SIM digital input/output port signals
PF[7:0]	Port F	SIM digital input/output port signals

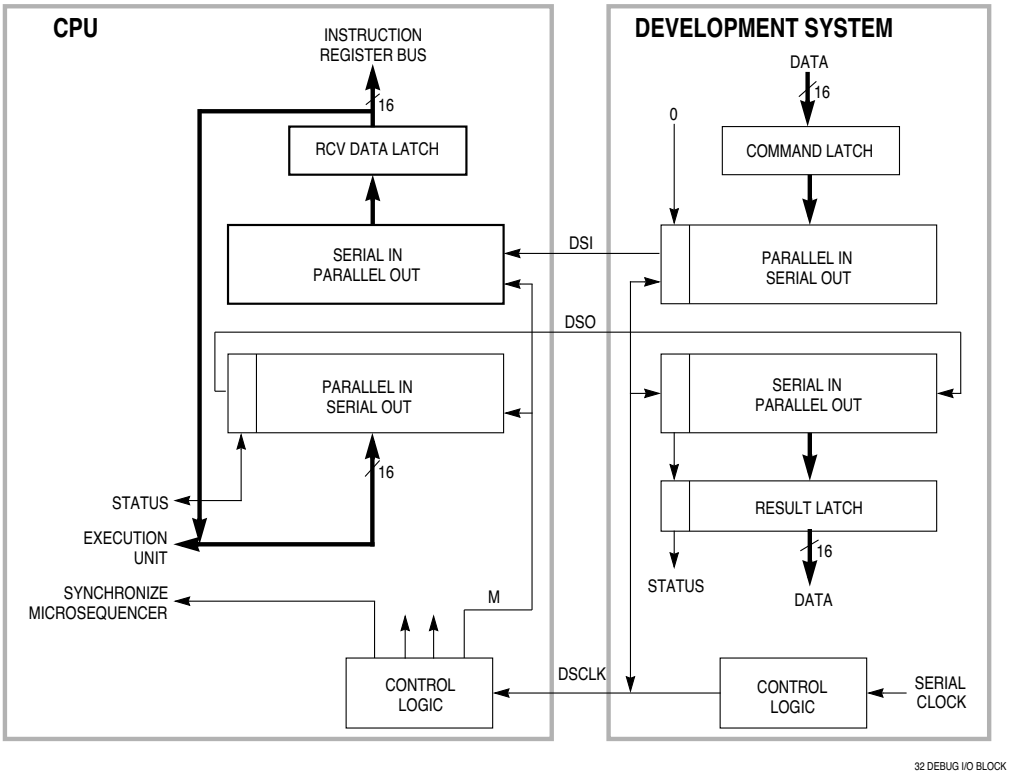

Figure 4-10 Debug Serial I/O Block Diagram

The serial interface uses a full-duplex synchronous protocol similar to the serial peripheral interface (SPI) protocol. The development system serves as the master of the serial link since it is responsible for the generation of DSCLK. If DSCLK is derived from the CPU32 system clock, development system serial logic is unhindered by the operating frequency of the target processor. Operable frequency range of the serial clock is from DC to one-half the processor system clock frequency.

The serial interface operates in full-duplex mode — data is transmitted and received simultaneously by both master and slave devices. In general, data transitions occur on the falling edge of DSCLK and are stable by the following rising edge of DSCLK. Data is transmitted MSB first, and is latched on the rising edge of DSCLK.

The serial data word is 17 bits wide, including 16 data bits and a status/control bit (refer to **Figure 4-11**). Bit 16 indicates the status of CPU-generated messages. **Table 4-7** shows the CPU-generated message types.

8.12.3.3 Continuous-Scan Modes

When application software requires execution of multiple passes through a sequence of conversions defined by a queue, a continuous-scan queue operating mode is selected.

When a queue is programmed for a continuous-scan mode, the single-scan enable bit in the queue control register does not have any meaning or effect. As soon as the queue operating mode is programmed, the selected trigger event can initiate queue execution.

In the case of the software initiated continuous-scan mode, the trigger event is generated internally and queue execution begins immediately. In the other continuous-scan queue operating modes, the selected trigger event must occur before the queue can start. A trigger overrun is recorded if a trigger event occurs during queue execution in the external trigger continuous-scan mode and the periodic timer continuous-scan mode. When a pause is encountered during a scan, another trigger event is required for queue execution to continue. Software involvement is not required for queue execution to continue from the paused state.

After queue execution is complete, the queue status is shown as idle. Since the continuous-scan queue operating modes allow an entire queue to be scanned multiple times, software involvement is not required for queue execution to continue from the idle state. The next trigger event causes queue execution to begin again, starting with the first CCW in the queue.

NOTE

It may not be possible to guarantee coherent samples when using the continuous-scan queue operating modes since the relationship between any two conversions may be variable due to programmable trigger events and queue priorities.

By programming the MQ1 field in QACR1 or the MQ2 field in QACR2, the following modes can be selected for queue 1 and/or 2:

- Software initiated continuous-scan mode
  - When this mode is programmed, the trigger event is generated automatically by the QADC, and queue execution begins immediately. If a pause is encountered, queue execution ceases for two QCLKs, while another trigger event is generated internally; execution then continues. When the end-of-queue is reached, another internal trigger event is generated, and queue execution begins again from the beginning of the queue.
  - While the time to internally generate and act on a trigger event is very short, software can momentarily read the status conditions, indicating that the queue is paused or idle. The trigger overrun flag is never set while in the software initiated continuous-scan mode.

arbitration priority. During arbitration, the BIUSM provides the arbitration value specified by IARB[2:0] in BIUMCR and IARB3 in FCSMSIC. If the CTM4 wins arbitration, it responds with a vector number generated by concatenating VECT[7:6] in BIUMCR and the six low-order bits specified by the number of the submodule requesting service. Thus, for FCSM12 in CTM4, six low-order bits would be 12 in decimal, or %001100 in binary.

10.6.6 FCSM Registers

The FCSM contains a status/interrupt/control register and a counter register. All unused bits and reserved address locations return zero when read. Writes to unused bits and reserved address locations have no effect. Refer to **D.7.6 FCSM Status/Interrupt/Control Register** and **D.7.7 FCSM Counter Register** for information concerning FCSM register and bit descriptions.

10.7 Modulus Counter Submodule (MCSM)

The modulus counter submodule (MCSM) is an enhanced FCSM. The MCSM contains a 16-bit modulus latch, a 16-bit loadable up-counter, counter loading logic, a clock selector, selectable time base bus drivers, and an interrupt interface. A modulus register provides the added flexibility of recycling the counter at a count other than 64K clock cycles. The content of the modulus latch is transferred to the counter when an overflow occurs, or when a user-specified edge transition occurs on a designated modulus load input pin. In addition, a write to the modulus counter simultaneously loads both the counter and the modulus latch with the specified value. The counter then begins incrementing from this new value.

In order to count, the MCSM requires the CPSM clock signals to be present. After reset, the MCSM does not count until the prescaler in the CPSM starts running (when the software sets the PRUN bit). This allows all counters in the CTM4 submodules to be synchronized.

The CTM4 contains two MCSMs. **Figure 10-4** shows a block diagram of the MCSM.

- Synchronized pulse width modulation
- Period measurement with additional transition detect
- Period measurement with missing transition detect
- Position-synchronized pulse generator
- Stepper motor
- Period/pulse width accumulator
- Quadrature decode

The G mask set (or motion control mask set) includes the following functions:

- Table stepper motor
- New input capture/transition counter
- Queued output match
- Programmable time accumulator
- Multichannel pulse width modulation
- Fast quadrature decode
- Universal asynchronous receiver/transmitter
- Brushless motor communication
- Frequency measurement
- Hall effect decode

11.2 TPU Components

The TPU consists of two 16-bit time bases, sixteen independent timer channels, a task scheduler, a microengine, and a host interface. In addition, a dual-ported parameter RAM is used to pass parameters between the module and the CPU32.

11.2.1 Time Bases

Two 16-bit counters provide reference time bases for all output compare and input capture events. Prescalers for both time bases are controlled by the CPU32 via bit fields in the TPU module configuration register (TPUMCR). Timer count registers TCR1 and TCR2 provide access to the current counter values. TCR1 and TCR2 can be read by TPU microcode, but are not directly available to the CPU32. The TCR1 clock is derived from the system clock. The TCR2 clock can be derived from the system clock or from an external clock input via the T2CLK pin.

11.2.2 Timer Channels

The TPU has 16 independent channels, each connected to an MCU pin. The channels have identical hardware. Each channel consists of an event register and pin control logic. The event register contains a 16-bit capture register, a 16-bit compare/match register, and a 16-bit greater-than-or-equal-to comparator. The direction of each pin, either output or input, is determined by the TPU microengine. Each channel can either use the same time base for match and capture, or can use one time base for match and the other for capture.

TRAMBAR can be written only once after a reset. This prevents runaway software from accidentally re-mapping the array. Because the locking mechanism is activated by the first write after a reset, the base address field should be written in a single word operation. Writing only one-half of the register prevents the other half from being written.

12.4 TPURAM Privilege Level

The RASP field in TRAMMCR specifies whether access to the TPURAM can be made from supervisor mode only, or from either user or supervisor mode. If supervisor-only access is specified, an access from user mode is ignored by the TPURAM control logic and can be decoded externally. Refer to **4.7 Privilege Levels** and **5.5.1.7 Function Codes** for more information concerning privilege levels.

12.5 Normal Operation

During normal operation, the TPURAM control registers and array can be accessed by the CPU32, by byte, word, or long word. A byte or aligned word access takes one bus cycle (two system clock cycles). A long word access requires two bus cycles. Misaligned accesses are not permitted by the CPU32 and will result in an address error exception. Refer to **5.6 Bus Operation** for more information concerning access times. The TPU cannot access the array and has no effect on the operation of the TPURAM during normal operation.

12.6 Standby Operation

Standby mode maintains the RAM array when the MCU main power supply is turned off.

Relative voltage levels of the V_{DD} and V_{STBY} pins determine whether the TPURAM is in standby mode. TPURAM circuitry switches to the standby power source when specified limits are exceeded. The TPURAM is essentially powered by the power supply pin with the greatest voltage (for example, V_{DD} or V_{STBY}). If specified standby supply voltage levels are maintained during the transition, there is no loss of memory when switching occurs. The RAM array cannot be accessed while the TPURAM is powered from V_{STBY}. If standby operation is not desired, connect the V_{STBY} pin to the V_{SS} pin.

I_{SB} (SRAM standby current) may exceed specified maximum standby current during the time V_{DD} makes the transition from normal operating level to the level specified for standby operation. This occurs within the voltage range $V_{SB} - 0.5\ V \geq V_{DD} \geq V_{SS} + 0.5\ V$. Typically, I_{SB} peaks when $V_{DD} \approx V_{SB} - 1.5\ V$, and averages 1.0 mA over the transition period.

Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for standby switching and power consumption specifications.

bit in the matching transmit message buffer is set, the TouCAN will transmit a remote frame as a response.

A received remote frame is not stored in a receive message buffer. It is only used to trigger the automatic transmission of a frame in response. The mask registers are not used in remote frame ID matching. All ID bits (except RTR) of the incoming received frame must match for the remote frame to trigger a response transmission.

13.5.6 Overload Frames

Overload frame transmissions are not initiated by the TouCAN unless certain conditions are detected on the CAN bus. These conditions include:

- Detection of a dominant bit in the first or second bit of intermission.
- Detection of a dominant bit in the seventh (last) bit of the end-of-frame (EOF) field in receive frames.
- Detection of a dominant bit in the eighth (last) bit of the error frame delimiter or overload frame delimiter.

13.6 Special Operating Modes

The TouCAN module has three special operating modes:

- Debug mode
- Low-power stop mode
- Auto power save mode

13.6.1 Debug Mode

Debug mode is entered by setting the HALT bit in the CANMCR, or by assertion of the IMB FREEZE line. In both cases, the FRZ1 bit in CANMCR must also be set to allow HALT or FREEZE to place the TouCAN in debug mode.

Once entry into debug mode is requested, the TouCAN waits until an intermission or idle condition exists on the CAN bus, or until the TouCAN enters the error passive or bus off state. Once one of these conditions exists, the TouCAN waits for the completion of all internal activity. When this happens, the following events occur:

- The TouCAN stops transmitting/receiving frames.
- The prescaler is disabled, thus halting all CAN bus communication.
- The TouCAN ignores its RX pins and drives its TX pins as recessive.
- The TouCAN loses synchronization with the CAN bus and the NOTRDY and FRZACK bits in CANMCR are set.
- The CPU32 is allowed to read and write the error counter registers.

After engaging one of the mechanisms to place the TouCAN in debug mode, the user must wait for the FRZACK bit to be set before accessing any other registers in the TouCAN, otherwise unpredictable operation may occur.

To exit debug mode, the IMB FREEZE line must be negated or the HALT bit in CANMCR must be cleared.

NOTES:

1. Applies to :
   Port E[7:4] — SIZ[1:0], $\overline{AS}$, $\overline{DS}$
   Port F[7:0] — $\overline{IRQ[7:1]}$, MODCLK
   Port QS[7:0] — TXD, PCS[3:1], PCS0/$\overline{SS}$, SCK, MOSI, MISO
   TPUCH[15:0], T2CLK, CPWM[8:5], CTD[4:3], CTD[10:9], CTM2C
   $\overline{BKPT}$/DSCLK, $\overline{IFETCH}$, $\overline{RESET}$, RXD, $\overline{TSTME}$/TSC
   EXTAL (when PLL enabled)
2. Input-Only Pins: EXTAL, $\overline{TSTME}$/TSC, $\overline{BKPT}$, PAI, T2CLK, RXD, CTM2C
   Output-Only Pins: $\overline{CSBOOT}$, $\overline{BG}$/$\overline{CS}$, CLKOUT, FREEZE/QUOT, $\overline{IPIPE}$
   Input/Output Pins:
   Group 1: DATA[15:0], $\overline{IFETCH}$, TPUCH[15:0], CPWM[8:5], CTD[4:3], CTD[10:9]
   Group 2: Port C[6:0] — ADDR[22:19]/$\overline{CS[9:6]}$, FC[2:0]/$\overline{CS[5:3]}$
   Port E[7:0] — SIZ[1:0], $\overline{AS}$, $\overline{DS}$, $\overline{AVEC}$, $\overline{RMC}$, $\overline{DSACK[1:0]}$
   Port F[7:0] — $\overline{IRQ[7:1]}$, MODCLK
   Port QS[7:3] — TXD, PCS[3:1], PCS0/$\overline{SS}$
   ADDR23/$\overline{CS10}$/ECLK, ADDR[18:0], R/$\overline{W}$, $\overline{BERR}$, $\overline{BR}$/$\overline{CS0}$, $\overline{BGACK}$/$\overline{CS2}$
   Group 3: $\overline{HALT}$, $\overline{RESET}$
   Group 4: MISO, MOSI, SCK
   Pin groups do not include QADC pins. See **Tables A-11** through **A-14** for information concerning the QADC.
3. Does not apply to $\overline{HALT}$ and $\overline{RESET}$ because they are open drain pins. Does not apply to port QS[7:0] (TXD, PCS[3:1], PCS0/$\overline{SS}$, SCK, MOSI, MISO) in wired-OR mode.
4. Use of an active pulldown device is recommended.
5. Total operating current is the sum of the appropriate I_{DD}, I_{DDSYN}, and I_{SB} values. I_{DD} values include supply currents for device modules powered by V_{DDE} and V_{DDI} pins.
6. Current measured at maximum system clock frequency, all modules active.
7. The SRAM module will not switch into standby mode as long as V_{SB} does not exceed V_{DD} by more than 0.5 volts. The SRAM array cannot be accessed while the module is in standby mode.
8. When V_{DD} is transitioning during power-up or power down sequence, and V_{SB} is applied, current flows between the V_{STBY} and V_{DD} pins, which causes standby current to increase toward the maximum transient condition specification. System noise on the V_{DD} and V_{STBY} pins can contribute to this condition.
9. Power dissipation measured at system clock frequency, all modules active. Power dissipation can be calculated using the following expression:
   $$P_D = \text{Maximum } V_{DD} (\text{Run } I_{DD} + I_{DDSYN} + I_{SB}) + \text{Maximum } V_{DDA} (I_{DDA})$$
10. This parameter is periodically sampled rather than 100% tested.

Table A-11 QADC Maximum Ratings

Num	Parameter	Symbol	Min	Max	Unit
1	Analog Supply, with reference to V_{SSA}	V_{DDA}	– 0.3	6.5	V
2	Internal Digital Supply, with reference to V_{SSI}	V_{DDI}	– 0.3	6.5	V
3	Reference Supply, with reference to V_{RL}	V_{RH}	– 0.3	6.5	V
4	V_{SS} Differential Voltage	$V_{SSI} – V_{SSA}$	– 0.1	0.1	V
5	V_{DD} Differential Voltage	$V_{DDI} – V_{DDA}$	– 6.5	6.5	V
6	V_{REF} Differential Voltage	$V_{RH} – V_{RL}$	– 6.5	6.5	V
7	V_{RH} to V_{DDA} Differential Voltage	$V_{RH} – V_{DDA}$	– 6.5	6.5	V
8	V_{RL} to V_{SSA} Differential Voltage	$V_{RL} – V_{SSA}$	– 6.5	6.5	V
9	Disruptive Input Current[1, 2, 3, 4, 5, 6, 7] $V_{NEGCLAMP}$ = – 0.3 V $V_{POSCLAMP}$ = 8 V	I_{NA}	– 500	500	µA
10	Positive Overvoltage Current Coupling Ratio[1, 5, 6, 8] PQA PQB	K_P	 2000 2000	—	—
11	Negative Overvoltage Current Coupling Ratio[1, 5, 6, 8] PQA PQB	K_N	 125 500	—	—
12	Maximum Input Current[3, 4, 6] $V_{NEGCLAMP}$ = – 0.3 V $V_{POSCLAMP}$ = 8 V	I_{MA}	– 25	25	mA

NOTES:

1. Below disruptive current conditions, the channel being stressed has conversion values of $3FF for analog inputs greater than V_{RH} and $000 for values less than V_{RL}. This assumes that $V_{RH} \leq V_{DDA}$ and $V_{RL} \geq V_{SSA}$ due to the presence of the sample amplifier. Other channels are not affected by non-disruptive conditions.
2. Input signals with large slew rates or high frequency noise components cannot be converted accurately. These signals also affect the conversion accuracy of other channels.
3. Exceeding limit may cause conversion error on stressed channels and on unstressed channels. Transitions within the limit do not affect device reliability or cause permanent damage.
4. Input must be current limited to the value specified. To determine the value of the required current-limiting resistor, calculate resistance values using positive and negative clamp values, then use the larger of the calculated values.
5. This parameter is periodically sampled rather 100% tested.
6. Condition applies to one pin at a time.
7. Determination of actual maximum disruptive input current, which can affect operation, is related to external system component values.
8. Current coupling is the ratio of the current induced from overvoltage (positive or negative, through an external series coupling resistor), divided by the current induced on adjacent pins. A voltage drop may occur across the external source impedances of the adjacent pins, impacting conversions on these adjacent pins.

D.2 System Integration Module

Table D-3 shows the SIM address map. The column labeled “Access” indicates the privilege level at which the CPU32 must be operating to access the register. A designation of “S” indicates that supervisor mode is required. A designation of “S/U” indicates that the register can be programmed for either supervisor mode access or unrestricted access.

Table D-3 SIM Address Map

Access	Address[1]	15 8	7 0
S	$YFFA00	SIM Module Configuration Register (SIMCR)	
S	$YFFA02	SIM Test Register (SIMTR)	
S	$YFFA04	Clock Synthesizer Control Register (SYNCR)	
S	$YFFA06	Not Used	Reset Status Register (RSR)
S	$YFFA08	SIM Test Register E (SIMTRE)	
S	$YFFA0A	Not Used	
S	$YFFA0C	Not Used	
S	$YFFA0E	Not Used	
S/U	$YFFA10	Not Used	Port E Data (PORTE0)
S/U	$YFFA12	Not Used	Port E Data (PORTE1)
S/U	$YFFA14	Not Used	Port E Data Direction (DDRE)
S	$YFFA16	Not Used	Port E Pin Assignment (PEPAR)
S/U	$YFFA18	Not Used	Port F Data (PORTF0)
S/U	$YFFA1A	Not Used	Port F Data (PORTF1)
S/U	$YFFA1C	Not Used	Port F Data Direction (DDRF)
S	$YFFA1E	Not Used	Port F Pin Assignment (PFPAR)
S	$YFFA20	Not Used	System Protection Control (SYPCR)
S	$YFFA22	Periodic Interrupt Control Register (PICR)	
S	$YFFA24	Periodic Interrupt Timing Register (PITR)	
S	$YFFA26	Not Used	Software Service (SWSR)
S	$YFFA28	Not Used	
S	$YFFA2A	Not Used	
S	$YFFA2C	Not Used	
S	$YFFA2E	Not Used	
S	$YFFA30	Test Module Master Shift A (TSTMSRA)	
S	$YFFA32	Test Module Master Shift B (TSTMSRB)	
S	$YFFA34	Test Module Shift Count (TSTSC)	
S	$YFFA36	Test Module Repetition Counter (TSTRC)	
S	$YFFA38	Test Module Control (CREG)	
S/U	$YFFA3A	Test Module Distributed (DREG)	
	$YFFA3C	Not Used	
	$YFFA3E	Not Used	
S/U	$YFFA40	Not Used	Port C Data (PORTC)
	$YFFA42	Not Used	
S	$YFFA44	Chip-Select Pin Assignment (CSPAR0)	
S	$YFFA46	Chip-Select Pin Assignment (CSPAR1)	
S	$YFFA48	Chip-Select Base Boot (CSBARBT)	
S	$YFFA4A	Chip-Select Option Boot (CSORBT)	
S	$YFFA4C	Chip-Select Base 0 (CSBAR0)	
S	$YFFA4E	Chip-Select Option 0 (CSOR0)	

To reset the software watchdog:

1. Write $55 to SWSR.
2. Write $AA to SWSR.

Both writes must occur in the order specified before the software watchdog times out, but any number of instructions can occur between the two writes.

D.2.16 Port C Data Register

PORTC — Port C Data Register **$YFFA41**

15 … 8	7	6	5	4	3	2	1	0
NOT USED	0	PC6	PC5	PC4	PC3	PC2	PC1	PC0
RESET:	0	1	1	1	1	1	1	1

PORTC latches data for chip-select pins configured as discrete outputs.

D.2.17 Chip-Select Pin Assignment Registers

CSPAR0 — Chip-Select Pin Assignment Register 0 **$YFFA44**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	CS5PA[1:0]		CS4PA[1:0]		CS3PA[1:0]		CS2PA[1:0]		CS1PA[1:0]		CS0PA[1:0]		CSBTPA[1:0]	
RESET:															
0	0	DATA2	1	DATA2	1	DATA2	1	DATA1	1	DATA1	1	DATA1	1	1	DATA0

The chip-select pin assignment registers configure the chip-select pins for use as discrete I/O, an alternate function, or as an 8-bit or 16-bit chip-select. Each 2-bit field in CSPAR[0:1] (except for CSBTPA[1:0]) has the possible encoding shown in **Table D-9**.

Table D-9 Pin Assignment Field Encoding

CSxPA[1:0]	Description
00	Discrete output[1]
01	Alternate function[1]
10	Chip-select (8-bit port)
11	Chip-select (16-bit port)

NOTES:
1. Does not apply to the $\overline{\text{CSBOOT}}$ field.

CSPAR0 contains seven 2-bit fields that determine the function of corresponding chip-select pins. Bits [15:14] are not used. These bits always read zero; writes have no effect. CSPAR0 bit 1 always reads one; writes to CSPAR0 bit 1 have no effect. The alternate functions can be enabled by data bus mode selection during reset.

Table D-10 shows CSPAR0 pin assignments.

DSCK — PCS to SCK Delay

0 = PCS valid to SCK delay is one-half SCK.

1 = SPCR1 DSCKL[6:0] specifies delay from PCS valid to SCK.

PCS[3:0] — Peripheral Chip Select

Use peripheral chip-select bits to select an external device for serial data transfer. More than one peripheral chip select may be activated at a time, and more than one peripheral chip can be connected to each PCS pin, provided proper fanout is observed. PCS0 shares a pin with the slave select ($\overline{SS}$) signal, which initiates slave mode serial transfer. If $\overline{SS}$ is taken low when the QSPI is in master mode, a mode fault occurs.

Table D-49 PWMSM Output Pin Polarity Selection

POL	EN	Output Pin State	Periodic Edge	Variable Edge	Optional Interrupt On
0	0	Always low	—	—	—
1	0	Always high	—	—	—
0	1	High pulse	Rising edge	Falling edge	Rising edge
1	1	Low pulse	Falling edge	Rising edge	Falling edge

EN — PWMSM Enable

This control bit enables and disables the PWMSM.

0 = Disable the PWMSM.
1 = Enable the PWMSM.

While the PWMSM is disabled (EN = 0):

- The output flip-flop is held in reset and the level on the output pin is set to one or zero according to the state of the POL bit.
- The PWMSM divide-by-256 prescaler is held in reset.
- The counter stops incrementing and is at $0001.
- The comparators are disabled.
- The PWMA1 and PWMB1 registers permanently transfer their contents to the buffer registers PWMA2 and PWMB2, respectively.

When the EN bit is changed from zero to one:

- The output flip-flop is set to start the first pulse.
- The PWMSM divide-by-256 prescaler is released.
- The counter is released and starts to increment from $0001.
- The FLAG bit is set to indicate that PWMA1 and PWMB1 can be updated with new values.

While EN is set, the PWMSM continuously generates a pulse width modulated output signal based on the data in PWMA2 and PWMB2 which are updated via PWMA1 and PWMB2 each time a period is completed.

NOTE

To prevent unwanted output waveform glitches when disabling the PWMSM, first write to PWMB1 to generate one period of 0% duty cycle, then clear EN.

CLK[2:0] — Clock Rate Selection

The CLK[2:0] bits select one of the eight counter clock sources coming from the PWMSM prescaler. These bits can be changed at any time. **Table D-50** shows the counter clock sources and rates in detail.

CH[15:0] — Encoded Host Sequence

The host sequence field selects the mode of operation for the time function selected on a given channel. The meaning of the host sequence bits depends on the time function specified.

D.8.9 Host Service Request Registers

HSSR0 — Host Service Request Register 0 **$YFFE18**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 15		CH 14		CH 13		CH 12		CH 11		CH 10		CH 9		CH 8	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HSSR1 — Host Service Request Register 1 **$YFFE1A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 7		CH 6		CH 5		CH 4		CH 3		CH 2		CH 1		CH 0	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH[15:0] — Encoded Type of Host Service

The host service request field selects the type of host service request for the time function selected on a given channel. The meaning of the host service request bits depends on the time function specified.

A host service request field cleared to %00 signals the host that service is completed by the microengine on that channel. The host can request service on a channel by writing the corresponding host service request field to one of three non-zero states. The CPU32 should monitor the host service request register until the TPU clears the service request to %00 before any parameters are changed or a new service request is issued to the channel.

D.8.10 Channel Priority Registers

CPR0 — Channel Priority Register 0 **$YFFE1C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 15		CH 14		CH 13		CH 12		CH 11		CH 10		CH 9		CH 8	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CPR1 — Channel Priority Register 1 **$YFFE1E**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 7		CH 6		CH 5		CH 4		CH 3		CH 2		CH 1		CH 0	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0 = The TouCAN is not in low-power stop mode and its clocks are running.
1 = The TouCAN has entered low-power stop mode and its clocks are stopped

IARB[3:0] — Interrupt Arbitration ID

The IARB field is used to arbitrate between simultaneous interrupt requests of the same priority. Each module that can generate interrupt requests must be assigned a unique, non-zero IARB field value.

D.10.2 TouCAN Test Configuration Register

CANTCR — TouCAN Test Configuration Register **$YFF082**

Used for factory test only.

D.10.3 TouCAN Interrupt Configuration Register

CANICR — TouCAN Interrupt Configuration Register **$YFF084**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					ILCAN[2:0]			IVBA[2:0]			RESERVED				
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

ILCAN[2:0] — Interrupt Request Level

When the TouCAN generates an interrupt request, ILCAN[2:0] determines which of the interrupt request signals is asserted. When a request is acknowledged, the TouCAN compares ILCAN[2:0] to a mask value supplied by the CPU32 to determine whether to respond. ILCAN[2:0] must have a value in the range of $0 (interrupts disabled) to $7 (highest priority).

IVBA[2:0] — Interrupt Vector Base Address

The interrupt vector base address specifies the high-order three bits of all the vector numbers generated by the different TouCAN interrupt sources.

NOTE

If the TouCAN issues an interrupt request after reset and before IVBA[2:0] is initialized, it will drive $0F as the “uninitialized” interrupt vector in response to a CPU32 interrupt acknowledge cycle, regardless of the specific event.

D.10.4 Control Register 0

CANCTRL0 — Control Register 0 **$YFF086**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOFF MSK	ERR MSK	RESERVED		RXMODE[1:0]		TXMODE[1:0]		CANCTRL1							
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

PSEG1[2:0] — Phase Buffer Segment 1

The PSEG1 field defines the length of phase buffer segment 1 in the bit time.

The valid programmed values are 0 through 7.

The length of phase buffer segment 1 is calculated as follows:

$$\text{Phase Buffer Segment 1} = (\text{PSEG1} + 1)\text{Time Quanta}$$

PSEG2 — Phase Buffer Segment 2

The PSEG2 field defines the length of phase buffer segment 2 in the bit time.

The valid programmed values are 0 through 7.

The length of phase buffer segment 2 is calculated as follows:

$$\text{Phase Buffer Segment 2} = (\text{PSEG2} + 1)\text{Time Quanta}$$

D.10.8 Free Running Timer

TIMER — Free Running Timer Register **$YFF08A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER															
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

The free running timer counter can be read and written by the CPU32. The timer starts from zero after reset, counts linearly to $FFFF, and wraps around.

The timer is clocked by the TouCAN bit-clock. During a message, it increments by one for each bit that is received or transmitted. When there is no message on the bus, it increments at the nominal bit rate.

The timer value is captured at the beginning of the identifier field of any frame on the CAN bus. The captured value is written into the “time stamp” entry in a message buffer after a successful reception/transmission of a message.