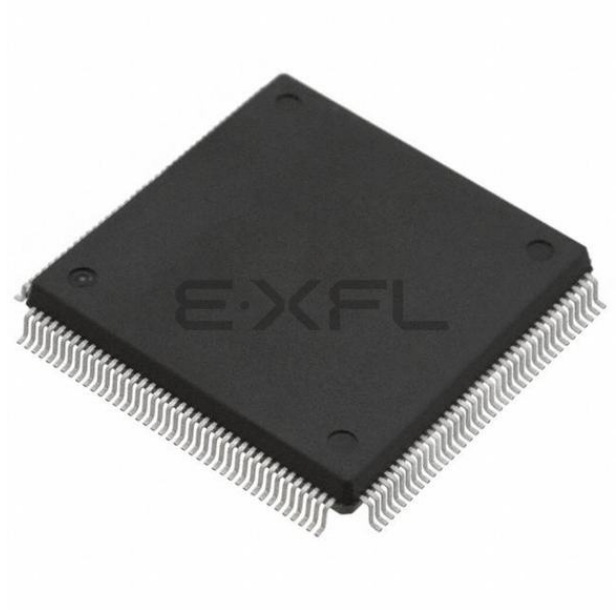

**Welcome to E-XFL.COM**

## 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 ~ 85°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/mc68336gcft20 |

## 2.4 Register Mnemonics

| | | |
|---|---|---|
| BIUMCR | — | CTM4 BIUSM Module Configuration |
| BIUTEST | — | CTM4 BIUSM Test Register |
| BIUTBR | — | CTM4 BIUSM Time Base Register |
| CANCTRL[0:2] | | TouCAN Control Register [0:2] |
| CANICR | — | TouCAN Interrupt Configuration Register |
| IFLAG | — | TouCAN Interrupt Flags Register |
| IMASK | — | TouCAN Interrupt Masks Register |
| CANMCR | — | TouCAN Module Configuration Register |
| CANTCR | — | TouCAN Test Configuration Register |
| CCW[0:27] | — | QADC Command Conversion Words [0:27] |
| CFSR[0:3] | — | TPU Channel Function Select Registers [0:3] |
| CIER | — | TPU Channel Interrupt Enable Register |
| CISR | — | TPU Channel Interrupt Status Register |
| CPCR | — | CTM4 CPSM Control Register |
| CPR[0:1] | — | TPU Channel Priority Registers [0:1] |
| CPTR | — | CTM4 CPSM Test Register |
| CR[0:F] | — | QSM Command RAM |
| CREG | — | SIM Test Control Register C |
| CSBARBT | — | SIM Chip-Select Base Address Register Boot ROM |
| CSBAR[0:10] | — | SIM Chip-Select Base Address Registers [0:10] |
| CSORBT | — | SIM Chip-Select Option Register Boot ROM |
| CSOR[0:10] | — | SIM Chip-Select Option Registers [0:10] |
| CSPAR[0:1] | — | SIM Chip-Select Pin Assignment Registers [0:1] |
| DASM[3:4]/[9:10]A | — | CTM4 DASM A Registers [3:4]/[9:10] |
| DASM[3:4]/[9:10]B | — | CTM4 DASM B Registers [3:4]/[9:10] |
| DASM[3:4]/[9:10]SIC | — | CTM4 DASM Status/Interrupt/Control Registers [3:4]/[9:10] |
| DCNR | — | Decoded Channel Number Register |
| DDRE | — | SIM Port E Data Direction Register |
| DDRF | — | SIM Port F Data Direction Register |
| DDRQA | — | QADC Port A Data Direction Register |
| DDRQS | — | QSM Port QS Data Direction Register |
| DREG | — | SIM Test Module Distributed Register |
| DSCR | — | TPU Development Support Control Register |
| DSSR | — | TPU Development Support Status Register |
| ESTAT | — | TouCAN Error and Status Register |

## 5.3 System Clock

The system clock in the SIM provides timing signals for the IMB modules and for an external peripheral bus. Because the MCU is a fully static design, register and memory contents are not affected when the clock rate changes. System hardware and software support changes in clock rate during operation.

The system clock signal can be generated from one of two sources. An internal phase-locked loop (PLL) can synthesize the clock from a fast reference, or the clock signal can be directly input from an external frequency source. The fast reference is typically a 4.194 MHz crystal, but may be generated by sources other than a crystal. Keep these sources in mind while reading the rest of this section. Refer to **Table A-4** in the **APPENDIX A ELECTRICAL CHARACTERISTICS** for clock specifications.

**Figure 5-2** is a block diagram of the clock submodule.

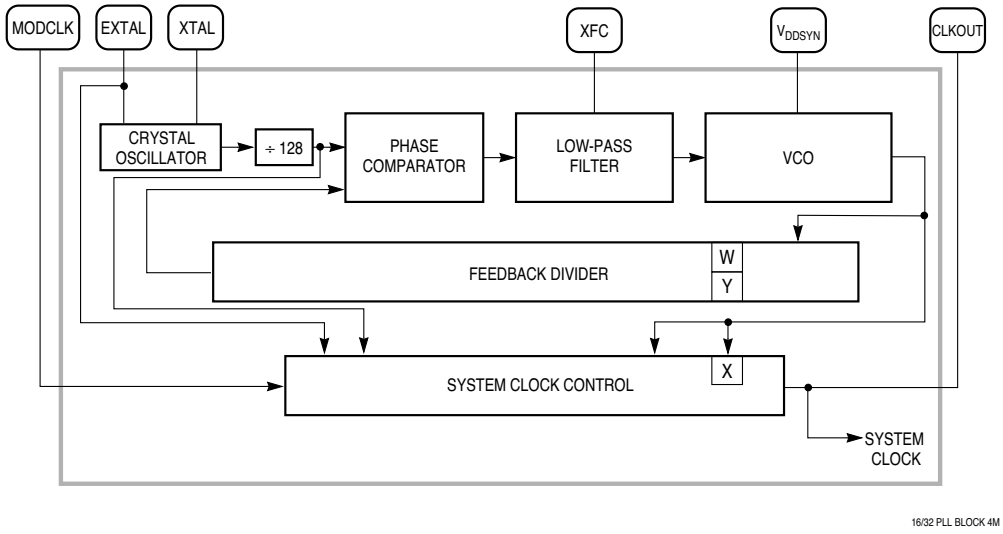


**Figure 5-2 System Clock Block Diagram**

### 5.3.1 Clock Sources

The state of the clock mode (MODCLK) pin during reset determines the system clock source. When MODCLK is held high during reset, the clock synthesizer generates a clock signal from an external reference frequency. The clock synthesizer control register (SYNCR) determines operating frequency and mode of operation. When MODCLK is held low during reset, the clock synthesizer is disabled and an external system clock signal must be driven onto the EXTAL pin.

The input clock is referred to as $f_{ref}$, and can be either a crystal or an external clock source. The output of the clock system is referred to as $f_{sys}$. Ensure that $f_{ref}$ and $f_{sys}$ are within normal operating limits.

### 5.3.3 External Bus Clock

The state of the E-clock division bit (EDIV) in SYNCR determines clock rate for the E-clock signal (ECLK) available on pin ADDR23. ECLK is a bus clock for M6800 devices and peripherals. ECLK frequency can be set to system clock frequency divided by eight or system clock frequency divided by sixteen. The clock is enabled by the $\overline{CS10}$ field in chip-select pin assignment register 1 (CSPAR1). ECLK operation during low-power stop is described in the following paragraph. Refer to **5.9 Chip-Selects** for more information about the external bus clock.

### 5.3.4 Low-Power Operation

Low-power operation is initiated by the CPU32. To reduce power consumption selectively, the CPU can set the STOP bits in each module configuration register. To minimize overall microcontroller power consumption, the CPU can execute the LPSTOP instruction, which causes the SIM to turn off the system clock.

When individual module STOP bits are set, clock signals inside each module are turned off, but module registers are still accessible.

When the CPU executes LPSTOP, a special CPU space bus cycle writes a copy of the current interrupt mask into the clock control logic. The SIM brings the MCU out of low-power stop mode when one of the following exceptions occur:

- $\overline{RESET}$
- Trace
- SIM interrupt of higher priority than the stored interrupt mask

Refer to **5.6.4.2 LPSTOP Broadcast Cycle** and **4.8.2.1 Low-Power Stop (LPSTOP)** for more information.

During low-power stop mode, unless the system clock signal is supplied by an external source and that source is removed, the SIM clock control logic and the SIM clock signal (SIMCLK) continue to operate. The periodic interrupt timer and input logic for the $\overline{RESET}$ and $\overline{IRQ}$ pins are clocked by SIMCLK, and can be used to bring the processor out of LPSTOP. Optionally, the SIM can also continue to generate the CLKOUT signal while in low-power stop mode.

STSIM and STEXT bits in SYNCR determine clock operation during low-power stop mode.

The flowchart shown in **Figure 5-5** summarizes the effects of the STSIM and STEXT bits when the MCU enters normal low power stop mode. Any clock in the off state is held low. If the synthesizer VCO is turned off during low-power stop mode, there is a PLL relock delay after the VCO is turned back on.

**Table 5-15 Reset Mode Selection**

| Mode Select Pin | Default Function (Pin Left High) | Alternate Function (Pin Pulled Low) |
|---|---|---|
| DATA0 | $\overline{\text{CSBOOT}}$ 16-bit | $\overline{\text{CSBOOT}}$ 8-bit |
| DATA1 | $\overline{\text{CS0}}$<br>$\overline{\text{CS1}}$<br>$\overline{\text{CS2}}$ | $\overline{\text{BR}}$<br>$\overline{\text{BG}}$<br>$\overline{\text{BGACK}}$ |
| DATA2 | $\overline{\text{CS3}}$<br>$\overline{\text{CS4}}$<br>$\overline{\text{CS5}}$ | FC0<br>FC1<br>FC2 |
| DATA3<br>DATA4<br>DATA5<br>DATA6<br>DATA7 | $\overline{\text{CS6}}$<br>$\overline{\text{CS}}[7:6]$<br>$\overline{\text{CS}}[8:6]$<br>$\overline{\text{CS}}[9:6]$<br>$\overline{\text{CS}}[10:6]$ | ADDR19<br>ADDR[20:19]<br>ADDR[21:19]<br>ADDR[22:19]<br>ADDR[23:19] |
| DATA8 | $\overline{\text{DSACK}}[1:0]$,<br>$\overline{\text{AVEC}}$, $\overline{\text{DS}}$, $\overline{\text{AS}}$, $\overline{\text{SIZ}}[1:0]$ | PORTE |
| DATA9 | $\overline{\text{IRQ}}[7:1]$<br>MODCLK | PORTF |
| DATA11 | Normal operation[1] | Reserved |
| MODCLK | VCO = System clock | EXTAL = System clock |
| $\overline{\text{BKPT}}$ | Background mode disabled | Background mode enabled |

NOTES:
1. The DATA11 bus must remain high during reset to ensure normal operation.

### 5.7.3.1 Data Bus Mode Selection

All data lines have weak internal pull-up drivers. When pins are held high by the internal drivers, the MCU uses a default operating configuration. However, specific lines can be held low externally during reset to achieve an alternate configuration.

**NOTE**

External bus loading can overcome the weak internal pull-up drivers on data bus lines and hold pins low during reset.

Use an active device to hold data bus lines low. Data bus configuration logic must release the bus before the first bus cycle after reset to prevent conflict with external memory devices. The first bus cycle occurs ten CLKOUT cycles after $\overline{\text{RESET}}$ is released. If external mode selection logic causes a conflict of this type, an isolation resistor on the driven lines may be required. **Figure 5-16** shows a recommended method for conditioning the mode select signals.

The mode configuration drivers are conditioned with R/$\overline{\text{W}}$ and $\overline{\text{DS}}$ to prevent conflicts between external devices and the MCU when reset is asserted. If external $\overline{\text{RESET}}$ is asserted during an external write cycle, R/$\overline{\text{W}}$ conditioning (as shown in **Figure 5-16**) prevents corruption of the data during the write. Similarly, $\overline{\text{DS}}$ conditions the mode configuration drivers so that external reads are not corrupted when $\overline{\text{RESET}}$ is asserted during an external read cycle.

Chip-select assertion can be synchronized with bus control signals to provide output enable, read/write strobe, or interrupt acknowledge signals. Chip-select logic can also generate $\overline{\text{DSACK}}$ and $\overline{\text{AVEC}}$ signals internally. A single $\overline{\text{DSACK}}$ generator is shared by all chip-selects. Each signal can also be synchronized with the ECLK signal available on ADDR23.

When a memory access occurs, chip-select logic compares address space type, address, type of access, transfer size, and interrupt priority (in the case of interrupt acknowledge) to parameters stored in chip-select registers. If all parameters match, the appropriate chip-select signal is asserted. Select signals are active low.

If a chip-select function is given the same address as a microcontroller module or an internal memory array, an access to that address goes to the module or array, and the chip-select signal is not asserted. The external address and data buses do not reflect the internal access.

All chip-select circuits are configured for operation out of reset. However, all chip-select signals except $\overline{\text{CSBOOT}}$ are disabled, and cannot be asserted until the BYTE[1:0] field in the corresponding option register is programmed to a non-zero value to select a transfer size. The chip-select option register must not be written until a base address has been written to a proper base address register. Alternate functions for chip-select pins are enabled if appropriate data bus pins are held low at the release of $\overline{\text{RESET}}$. Refer to **5.7.3.1 Data Bus Mode Selection** for more information. **Figure 5-20** is a functional diagram of a single chip-select circuit.

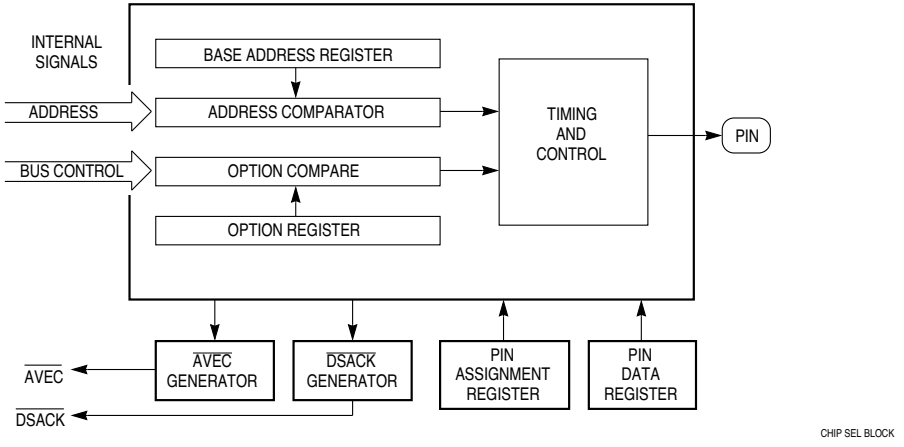


**Figure 5-20 Chip-Select Circuit Block Diagram**

## 8.9 External Multiplexing Operation

External multiplexers concentrate a number of analog signals onto a few inputs to the analog converter. This is helpful in applications that need to convert more analog signals than the A/D converter can normally provide. External multiplexing also puts the multiplexer closer to the signal source. This minimizes the number of analog signals that need to be shielded due to the close proximity of noisy, high speed digital signals near the MCU.

The QADC can use from one to four external multiplexers to expand the number of analog signals that may be converted. Up to 32 analog channels can be converted through external multiplexer selection. The externally multiplexed channels are automatically selected from the channel field of the conversion command word (CCW) table, the same as internally multiplexed channels.

All of the automatic queue features are available for externally and internally multiplexed channels. The software selects externally multiplexed mode by setting the MUX bit in QACR0.

**Figure 8-3** shows the maximum configuration of four external multiplexers connected to the QADC. The external multiplexers select one of eight analog inputs and connect it to one analog output, which becomes an input to the QADC. The QADC provides three multiplexed address signals (MA[2:0]), to select one of eight inputs. These outputs are connected to all four multiplexers. The analog output of each multiplexer is each connected to one of four separate QADC inputs — ANw, ANx, ANy, and ANz.

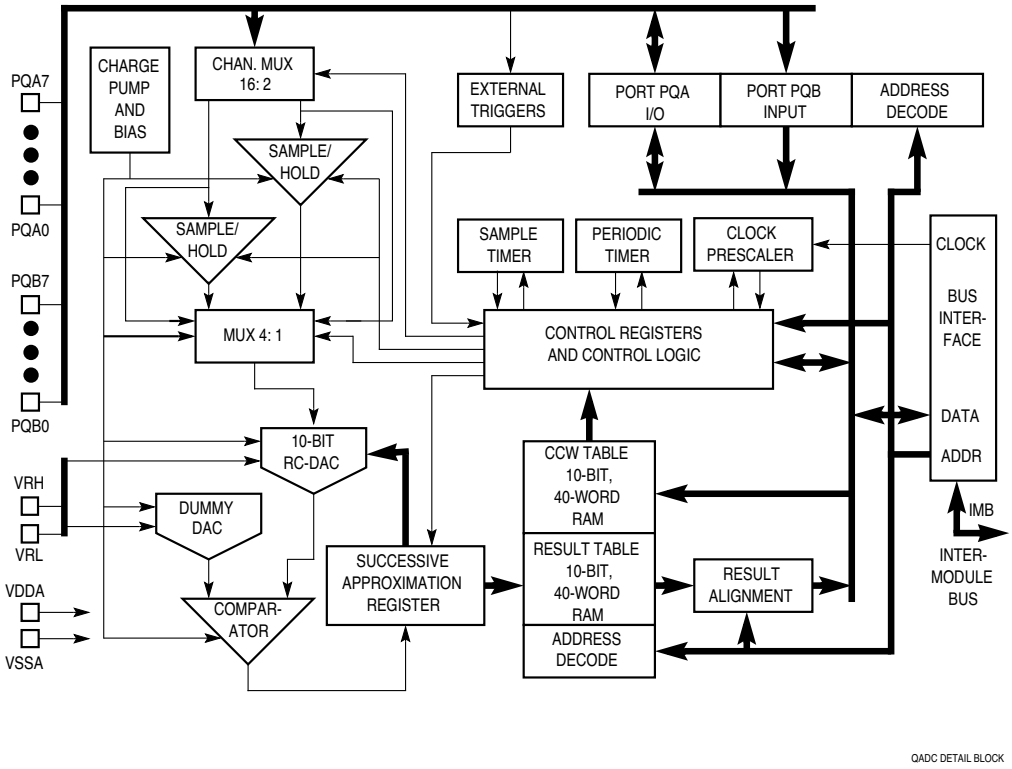


**Figure 8-4 QADC Module Block Diagram**

### 8.11.1 Conversion Cycle Times

Total conversion time is made up of initial sample time, transfer time, final sample time, and resolution time. Initial sample time refers to the time during which the selected input channel is connected to the sample capacitor at the input of the sample buffer amplifier. During the transfer period, the sample capacitor is disconnected from the multiplexer, and the stored voltage is buffered and transferred to the RC DAC array. During the final sampling period, the sample capacitor and amplifier are bypassed, and the multiplexer input charges the RC DAC array directly. During the resolution period, the voltage in the RC DAC array is converted to a digital value and stored in the SAR.

Initial sample time is fixed at two QCLKs and the transfer time at four QCLKs. Final sample time can be 2, 4, 8, or 16 ADC clock cycles, depending on the value of the IST field in the CCW. Resolution time is ten cycles.

Transfer and resolution require a minimum of 18 QCLK clocks (8.6 µs with a 2.1 MHz QCLK). If the maximum final sample time period of 16 QCLKs is selected, the total conversion time is 15.2 µs with a 2.1 MHz QCLK.

multiplication operation resolves down to the desired degrees. Two modes of operation allow pulse length to be determined either by angular position or by time.

Refer to TPU programming note *Position-Synchronized Pulse Generator (PSP) TPU Function* (TPUPN14/D) for more information.

### 11.4.9 Stepper Motor (SM)

The stepper motor control algorithm provides for linear acceleration and deceleration control of a stepper motor with a programmable number of step rates of up to 14. Any group of channels, up to eight, can be programmed to generate the control logic necessary to drive a stepper motor.

The time period between steps (P) is defined as:

$$P(r) = K1 - K2 \cdot r$$

where r is the current step rate (1–14), and K1 and K2 are supplied as parameters.

After providing the desired step position in a 16-bit parameter, the CPU32 issues a step request. Next, the TPU steps the motor to the desired position through an acceleration/deceleration profile defined by parameters. The parameter indicating the desired position can be changed by the CPU32 while the TPU is stepping the motor. This algorithm changes the control state every time a new step command is received.

A 16-bit parameter initialized by the CPU32 for each channel defines the output state of the associated pin. The bit pattern written by the CPU32 defines the method of stepping, such as full stepping or half stepping. With each transition, the 16-bit parameter rotates one bit. The period of each transition is defined by the programmed step rate.

Refer to TPU programming note *Stepper Motor (SM) TPU Function* (TPUPN13/D) for more information.

### 11.4.10 Period/Pulse-Width Accumulator (PPWA)

The period/pulse-width accumulator algorithm accumulates a 16-bit or 24-bit sum of either the period or the pulse width of an input signal over a programmable number of periods or pulses (from one to 255). After an accumulation period, the algorithm can generate a link to a sequential block of up to eight channels. The user specifies a starting channel of the block and number of channels within the block. Generation of links depends on the mode of operation. Any channel can be used to measure an accumulated number of periods of an input signal. A maximum of 24 bits can be used for the accumulation parameter. From one to 255 period measurements can be made and summed with the previous measurement(s) before the TPU interrupts the CPU, allowing instantaneous or average frequency measurement, and the latest complete accumulation (over the programmed number of periods).

The pulse width (high-time portion) of an input signal can be measured (up to 24 bits) and added to a previous measurement over a programmable number of periods (one

## 13.2 External Pins

The TouCAN module interface to the CAN bus is composed of four pins: CANTX0 and CANTX1, which transmit serial data, and CANRX0 and CANRX1, which receive serial data. **Figure 13-2** shows a typical CAN system.

**NOTE**

Pins CANTX1 and CANRX1 are not used on the MC68376.

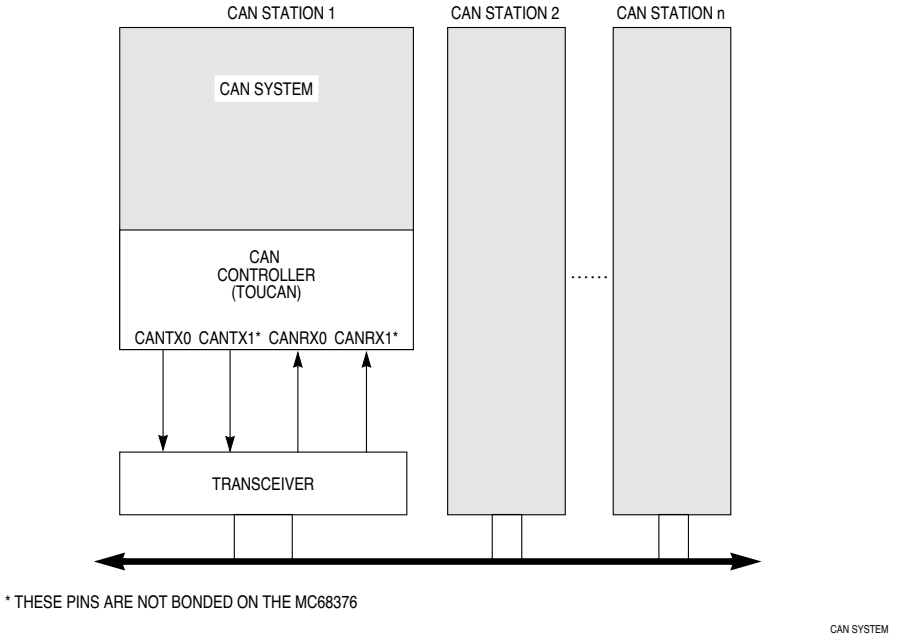


**Figure 13-2 Typical CAN Network**

Each CAN station is connected physically to the CAN bus through a transceiver. The transceiver provides the transmit drive, waveshaping, and receive/compare functions required for communicating on the CAN bus. It can also provide protection against damage to the TouCAN caused by a defective CAN bus or a defective CAN station.

## 13.3 Programmer's Model

The TouCAN module address space is split into 128 bytes starting at the base address, and then an extra 256 bytes starting at the base address +128. The upper 256 are fully used for the message buffer structures. Out of the lower 128 bytes, only part is occupied by various registers. Refer to **D.10 TouCAN Module** for detailed information on the TouCAN address map and register structure.

- Cascade usage of TX error counter with an additional internal counter to detect the 128 occurrences of 11 consecutive recessive bits necessary to transition from bus off into error active.

Both counters are read only (except in test/freeze/halt modes).

The TouCAN responds to any bus state as described in the CAN protocol, transmitting an error active or error passive flag, delaying its transmission start time (error passive) and avoiding any influence on the bus when in the bus off state. The following are the basic rules for TouCAN bus state transitions:

- If the value of the TX error counter or RX error counter increments to a value greater than or equal to 128, the fault confinement state (FCS[1:0]) field in the error status register is updated to reflect an error passive state.
- If the TouCAN is in an error passive state, and either the TX error counter or RX error counter decrements to a value less than or equal to 127, while the other error counter already satisfies this condition, the FCS[1:0] field in the error status register is updated to reflect an error active state.
- If the value of the TX error counter increases to a value greater than 255, the FCS[1:0] field in the error status register is updated to reflect a bus off state, and an interrupt may be issued. The value of the TX error counter is reset to zero.
- If the TouCAN is in the bus off state, the TX error counter and an additional internal counter are cascaded to count 128 occurrences of 11 consecutive recessive bits on the bus. To do this, the TX error counter is first reset to zero, then the internal counter begins counting consecutive recessive bits. Each time the internal counter counts 11 consecutive recessive bits, the TX error counter is incremented by one and the internal counter is reset to zero. When the TX error counter reaches the value of 128, the FCS[1:0] field in the error status register is updated to be error active, and both error counters are reset to zero. Any time a dominant bit is detected following a stream of less than 11 consecutive recessive bits, the internal counter resets itself to zero, but does not affect the TX error counter value.
- If only one node is operating in a system, the TX error counter will increment with each message it attempts to transmit, due to the resulting acknowledgment errors. However, acknowledgment errors will never cause the TouCAN to transition from the error passive state to the bus off state.
- If the RX error counter increments to a value greater than 127, it will stop incrementing, even if more errors are detected while being a receiver. After the next successful message reception, the counter is reset to a value between 119 and 127, to enable a return to the error active state.

### 13.4.5 Time Stamp

The value of the free-running 16-bit timer is sampled at the beginning of the identifier field on the CAN bus. For a message being received, the time stamp will be stored in the time stamp entry of the receive message buffer at the time the message is written into that buffer. For a message being transmitted, the time stamp entry will be written into the transmit message buffer once the transmission has completed successfully.

- If both STOP and SELFWAKE are set and a recessive to dominant edge immediately occurs on the CAN bus, the TouCAN may never set the STOPACK bit, and the STOP bit will be cleared.
- To prevent old frames from being sent when the TouCAN awakes from low-power stop mode via the self-wake mechanism, disable all transmit sources, including transmit buffers configured for remote request responses, before placing the TouCAN in low-power stop mode.
- If the TouCAN is in debug mode when the STOP bit is set, the TouCAN will assume that debug mode should be exited. As a result, it will try to synchronize with the CAN bus, and only then will it await the conditions required for entry into low-power stop mode.
- Unlike other modules, the TouCAN does not come out of reset in low-power stop mode. The basic TouCAN initialization procedure (see **13.5.2 TouCAN Initialization**) should be executed before placing the module in low-power stop mode.
- If the TouCAN is in low-power stop mode with the self-wake mechanism engaged and is operating with a single system clock per time quantum, there can be extreme cases in which TouCAN wake-up on recessive to dominant edge may not conform to the CAN protocol. TouCAN synchronization will be shifted one time quantum from the wake-up event. This shift lasts until the next recessive to dominant edge, which resynchronizes the TouCAN to be in conformance with the CAN protocol. The same holds true when the TouCAN is in auto power save mode and awakens on a recessive to dominant edge.

### 13.6.3 Auto Power Save Mode

Auto power save mode enables normal operation with optimized power savings. Once the auto power save (APS) bit in CANMCR is set, the TouCAN looks for a set of conditions in which there is no need for the clocks to be running. If these conditions are met, the TouCAN stops its clocks, thus saving power. The following conditions will activate auto power save mode.

- No RX/TX frame in progress.
- No transfer of RX/TX frames to and from a serial message buffer, and no TX frame awaiting transmission in any message buffer.
- No CPU32 access to the TouCAN module.
- The TouCAN is not in debug mode, low-power stop mode, or the bus off state.

While its clocks are stopped, if the TouCAN senses that any one of the aforementioned conditions is no longer true, it restarts its clocks. The TouCAN then continues to monitor these conditions and stops/restarts its clocks accordingly.

## Table A-2 Typical Ratings

| Num | Rating | Symbol | Value | Unit |
|---|---|---|---|---|
| 1 | Supply Voltage | $V_{DD}$ | 5.0 | V |
| 2 | Operating Temperature | $T_A$ | 25 | °C |
| 3 | $V_{DD}$ Supply Current<br>RUN<br>LPSTOP, VCO off<br>LPSTOP, External clock, maxi $f_{sys}$ | $I_{DD}$ | <br>113<br>125<br>3.75 | <br>mA<br>µA<br>mA |
| 4 | Clock Synthesizer Operating Voltage | $V_{DDSYN}$ | 5.0 | V |
| 5 | $V_{DDSYN}$ Supply Current<br>VCO on, maximum $f_{sys}$<br>External Clock, maximum $f_{sys}$<br>LPSTOP, VCO off<br>$V_{DD}$ powered down | $I_{DDSYN}$ | <br>1.0<br>5.0<br>100<br>50 | <br>mA<br>mA<br>µA<br>µA |
| 6 | RAM Standby Voltage | $V_{SB}$ | 3.0 | V |
| 7 | RAM Standby Current<br>Normal RAM operation<br>Standby operation | $I_{SB}$ | <br>7.0<br>40 | <br>µA<br>µA |
| 8 | Power Dissipation | $P_D$ | 570 | mW |

## Table A-3 Thermal Characteristics

| Num | Rating | Symbol | Value | Unit |
|---|---|---|---|---|
| 1 | Thermal Resistance<br>Plastic 160-Pin Surface Mount | $\theta_{JA}$ | 37 | °C/W |

The average chip-junction temperature (TJ) in C can be obtained from:

$$T_J = T_A + (P_D \times \Theta_{JA}) \quad (1)$$

where:

$T_A$ = Ambient Temperature, °C

$\Theta_{JA}$ = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D$ = $P_{INT} + P_{I/O}$

$P_{INT}$ = $I_{DD} \times V_{DD}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected. An approximate relationship between $P_D$ and $T_J$ (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D + (T_A + 273°C) + \Theta_{JA} \times P_{D^2} \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring $P_D$ (at equilibrium) for a known $T_A$. Using this value of K, the values of $P_D$ and $T_J$ can be obtained by solving equations (1) and (2) iteratively for any value of $T_A$.

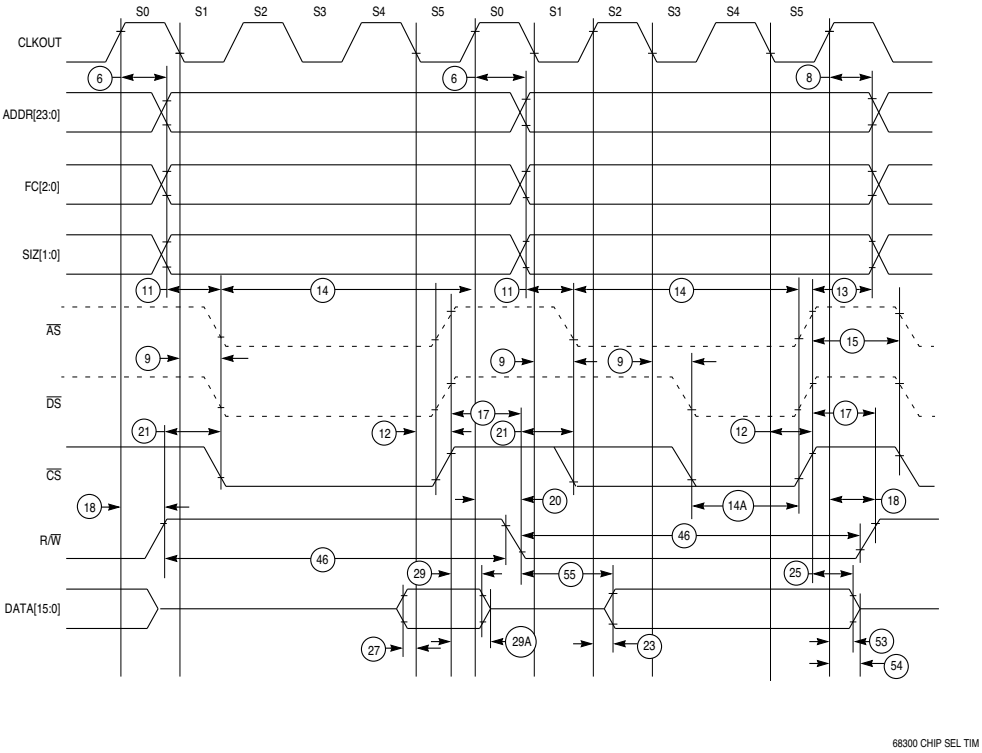


**Figure A-11 Chip-Select Timing Diagram**

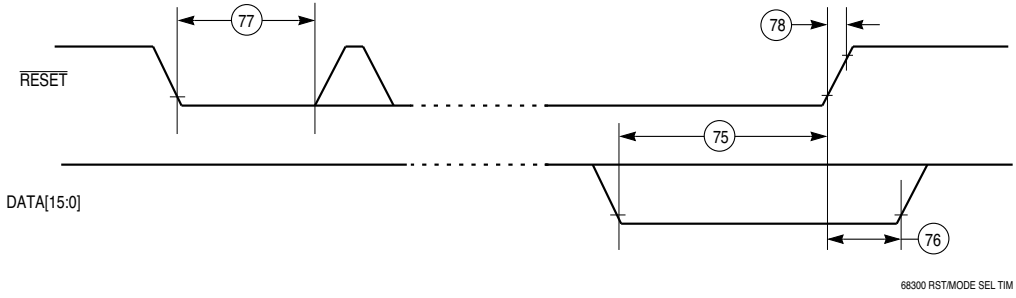


**Figure A-12 Reset and Mode Select Timing Diagram**

## B.1 Obtaining Updated MC68336/376 Mechanical Information

Although all devices manufactured by Motorola conform to current JEDEC standards, complete mechanical information regarding MC68336/376 microcontrollers is available through Motorola's website at motorola.com

To download updated package specifications, go to website

## B.2 Ordering Information

Refer to **Table B-1** for MC68336 ordering information and **Table B-2** for MC68376 ordering information. Contact a Motorola sales representative for information on ordering a custom ROM device.

**Table B-1 MC68336 Ordering Information**

| Part Number | Package Type | Frequency (MHz) | TPU | Temperature | Package Order Quantity | Order Number |
|---|---|---|---|---|---|---|
| MC68336 | 160-pin QFP | 20.97 MHz | A | –40 to +85 °C | 2 | SPMC68336ACFT20 |
| | | | | | 24 | MC68336ACFT20 |
| | | | | | 120 | MC68336ACFT20B1 |
| | | | | –40 to +105 °C | 2 | SPMC68336AVFT20 |
| | | | | | 24 | MC68336AVFT20 |
| | | | | | 120 | MC68336AVFT20B1 |
| | | | | –40 to +125 °C | 2 | SPMC68336AMFT20 |
| | | | | | 24 | MC68336AMFT20 |
| | | | | | 120 | MC68336AMFT20B1 |
| | | | G | –40 to +85 °C | 2 | SPMC68336GCFT20 |
| | | | | | 24 | MC68336GCFT20 |
| | | | | | 120 | MC68336GCFT20B1 |
| | | | | –40 to +105 °C | 2 | SPMC68336GVFT20 |
| | | | | | 24 | MC68336GVFT20 |
| | | | | | 120 | MC68336GVFT20B1 |
| | | | | –40 to +125 °C | 2 | SPMC68336GMFT20 |
| | | | | | 24 | MC68336GMFT20 |
| | | | | | 120 | MC68336GMFT20B1 |

**Table D-26 Queue 2 Operating Modes**

| MQ2[4:0] | Queue 2 Operating Mode |
|---|---|
| 00000 | Disabled mode, conversions do not occur |
| 00001 | Software triggered single-scan mode (started with SSE2) |
| 00010 | External trigger rising edge single-scan mode (on ETRIG2 pin) |
| 00011 | External trigger falling edge single-scan mode (on ETRIG2 pin) |
| 00100 | Interval timer single-scan mode: interval = QCLK period x $2^7$ |
| 00101 | Interval timer single-scan mode: interval = QCLK period x $2^8$ |
| 00110 | Interval timer single-scan mode: interval = QCLK period x $2^9$ |
| 00111 | Interval timer single-scan mode: interval = QCLK period x $2^{10}$ |
| 01000 | Interval timer single-scan mode: interval = QCLK period x $2^{11}$ |
| 01001 | Interval timer single-scan mode: interval = QCLK period x $2^{12}$ |
| 01010 | Interval timer single-scan mode: interval = QCLK period x $2^{13}$ |
| 01011 | Interval timer single-scan mode: interval = QCLK period x $2^{14}$ |
| 01100 | Interval timer single-scan mode: interval = QCLK period x $2^{15}$ |
| 01101 | Interval timer single-scan mode: interval = QCLK period x $2^{16}$ |
| 01110 | Interval timer single-scan mode: interval = QCLK period x $2^{17}$ |
| 01111 | Reserved mode |
| 10000 | Reserved mode |
| 10001 | Software triggered continuous-scan mode (started with SSE2) |
| 10010 | External trigger rising edge continuous-scan mode (on ETRIG2 pin) |
| 10011 | External trigger falling edge continuous-scan mode (on ETRIG2 pin) |
| 10100 | Periodic timer continuous-scan mode: period = QCLK period x $2^7$ |
| 10101 | Periodic timer continuous-scan mode: period = QCLK period x $2^8$ |
| 10110 | Periodic timer continuous-scan mode: period = QCLK period x $2^9$ |
| 10111 | Periodic timer continuous-scan mode: period = QCLK period x $2^{10}$ |
| 11000 | Periodic timer continuous-scan mode: period = QCLK period x $2^{11}$ |
| 11001 | Periodic timer continuous-scan mode: period = QCLK period x $2^{12}$ |
| 11010 | Periodic timer continuous-scan mode: period = QCLK period x $2^{13}$ |
| 11011 | Periodic timer continuous-scan mode: period = QCLK period x $2^{14}$ |
| 11100 | Periodic timer continuous-scan mode: period = QCLK period x $2^{15}$ |
| 11101 | Periodic timer continuous-scan mode: period = QCLK period x $2^{16}$ |
| 11110 | Periodic timer continuous-scan mode: period = QCLK period x $2^{17}$ |
| 11111 | Reserved mode |

RES — Queue 2 Resume

RES selects the resumption point after queue 2 is suspended by queue 1. If RES is changed during execution of queue 2, the change is not recognized until an end-of-queue condition is reached, or the queue operating mode of queue 2 is changed.

0 = After suspension, begin execution with the first CCW in queue 2 or the current subqueue.

1 = After suspension, begin execution with the aborted CCW in queue 2.

### D.5.8 Conversion Command Word Table

**CCW[0:27] —** Conversion Command Word Table **$YFF230–$YFF27E**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NOT USED | | | | | | P | BYP | IST[1:0] | | CHAN[5:0] | | | | | |
| RESET: | | | | | | U | U | U | U | U | U | U | U | U | U |

P — Pause

The pause bit allows the creation of sub-queues within queue 1 and queue 2. The QADC performs the conversion specified by the CCW with the pause bit set, and then the queue enters the pause state. Another trigger event causes execution to continue from the pause to the next CCW.

0 = Do not enter the pause state after execution of the current CCW.
1 = Enter the pause state after execution of the current CCW.

BYP — Sample Amplifier Bypass

Setting BYP enables the amplifier bypass mode for a conversion, and subsequently changes the timing. Refer to **8.11.1.1 Amplifier Bypass Mode Conversion Timing** for more information.

0 = Amplifier bypass mode disabled.
1 = Amplifier bypass mode enabled.

IST[1:0] — Input Sample Time

The IST field specifies the length of the sample window. Longer sample times permit more accurate A/D conversions of signals with higher source impedances.

**Table D-28** shows the bit encoding of the IST field.

**Table D-28 Input Sample Times**

| IST[1:0] | Input Sample Times |
|---|---|
| 00 | 2 QCLK periods |
| 01 | 4 QCLK periods |
| 10 | 8 QCLK periods |
| 11 | 16 QCLK periods |

CHAN[5:0] — Channel Number

The CHAN field selects the input channel number corresponding to the analog input pin to be sampled and converted. The analog input pin channel number assignments and the pin definitions vary depending on whether the QADC is operating in multiplexed or non-multiplexed mode. The queue scan mechanism sees no distinction between an internally or externally multiplexed analog input.

CHAN specifies a reserved channel number (channels 32 to 47) or an invalid channel number (channels 4 to 31 in non-multiplexed mode), the low reference level ($V_{RL}$) is converted. Programming the channel field to channel 63 indicates the end of the queue. Channels 60 to 62 are special internal channels. When one of these channels is selected, the sample amplifier is not used. The value of $V_{RL}$, $V_{RH}$, or $V_{DDA}/2$ is placed directly onto the converter. Programming the input sample time to any value other than two for one of the internal channels has no benefit except to lengthen the overall conversion time.

DTL[7:0] — Length of Delay after Transfer

When the DT bit is set in a command RAM byte, this field determines the length of the delay after a serial transfer. The following equation is used to calculate the delay:

$$\text{Delay after Transfer} = \frac{32 \times \text{DTL[7:0]}}{\text{System Clock}}$$

where DTL equals is in the range of 1 to 255.
A zero value for DTL[7:0] causes a delay-after-transfer value of 8192 ÷ $f_{sys}$.
If DT is zero in a command RAM byte, a standard delay is inserted.

$$\text{Standard Delay after Transfer} = \frac{17}{f_{sys}}$$

Delay after transfer can be used to provide a peripheral deselect interval. A delay can also be inserted between consecutive transfers to allow serial A/D converters to complete conversion.

### D.6.13 QSPI Control Register 2

**SPCR2 —** QSPI Control Register 2 **$YFFC1C**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SPIFIE | WREN | WRTO | 0 | ENDQP[3:0] | | | | 0 | 0 | 0 | 0 | NEWQP[3:0] | | | |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SPCR2 contains QSPI queue pointers, wraparound mode control bits, and an interrupt enable bit. The CPU32 has read/write access to SPCR2, but the QSM has read access only. SPCR2 is buffered. New SPCR2 values become effective only after completion of the current serial transfer. Rewriting NEWQP in SPCR2 causes execution to restart at the designated location. Reads of SPCR2 return the value of the register, not the buffer.

SPIFIE — SPI Finished Interrupt Enable
- 0 = QSPI interrupts disabled.
- 1 = QSPI interrupts enabled.

WREN — Wrap Enable
- 0 = Wraparound mode disabled.
- 1 = Wraparound mode enabled.

WRTO — Wrap To
- 0 = Wrap to pointer address $0.
- 1 = Wrap to address in NEWQP.

Bit 12 — Not Implemented

When the interrupt level set specified by IL[2:0] is non-zero, an interrupt request is generated when the FLAG bit is set.

IL[2:0] — Interrupt Level Field

When the PWMSM generates an interrupt request, IL[2:0] determines which of the interrupt request signals is asserted. When a request is acknowledged, the CTM4 compares IL[2:0] to a mask value supplied by the CPU32 to determine whether to respond. IL[2:0] must have a value in the range of $0 (interrupts disabled) to $7 (highest priority).

IARB3 — Interrupt Arbitration Bit 3

This bit and the IARB[2:0] field in BIUMCR are concatenated to determine the interrupt arbitration number for the submodule requesting interrupt service. Refer to **D.7.1 BIU Module Configuration Register** for more information on IARB[2:0].

PIN — Output Pin Status

This status bit indicates the logic state present on the PWM output pin.

0 = Logic zero present on the PWM output pin.
1 = Logic one present on the PWM output pin.

PIN is a read-only bit; writing to it has no effect.

LOAD — Period and Pulse Width Register Load Control

Setting LOAD reinitializes the PWMSM and starts a new PWM period without causing a glitch on the output signal.

0 = No action
1 = Load period and pulse width registers

This bit is always read as a zero. Writing a one to this bit results in the following immediate actions:

- The contents of PWMA1 (period value) are transferred to PWMA2.
- The contents of PWMB1 (pulse width value) are transferred to PWMB2.
- The counter register (PWMC) is initialized to $0001.
- The control logic and state sequencer are reset.
- The FLAG bit is set.
- The output flip-flop is set if the new value in PWMB2 is not $0000.

**NOTE**

Writing a one to the LOAD bit when the EN bit = 0, (when the PWMSM is disabled), has no effect.

POL — Output Pin Polarity Control

This control bit sets the polarity of the PWM output signal. It works in conjunction with the EN bit and controls whether the PWMSM drives the output pin with the non-inverted or inverted state of the output flip-flop. Refer to **Table D-49**.

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.

## –Q–