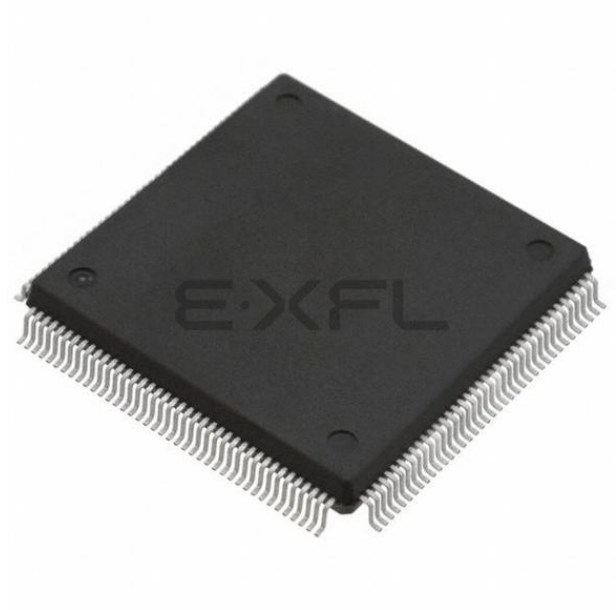


Welcome to E-XFL.COM

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	25MHz
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/mc68376bavab25

SECTION 3 OVERVIEW

This section contains information about the entire MC68336/376 modular microcontroller. It lists the features of each module, shows device functional divisions and pin assignments, summarizes signal and pin functions, discusses the intermodule bus, and provides system memory maps. Timing and electrical specifications for the entire microcontroller and for individual modules are provided in **APPENDIX A ELECTRICAL CHARACTERISTICS**. Comprehensive module register descriptions and memory maps are provided in **APPENDIX D REGISTER SUMMARY**.

3.1 MCU Features

The following paragraphs highlight capabilities of each of the microcontroller modules. Each module is discussed separately in a subsequent section of this user's manual.

3.1.1 Central Processing Unit (CPU32)

- 32-bit architecture
- Virtual memory implementation
- Table look-up and interpolate instruction
- Improved exception handling for controller applications
- High level language support
- Background debug mode
- Fully static operation

3.1.2 System Integration Module (SIM)

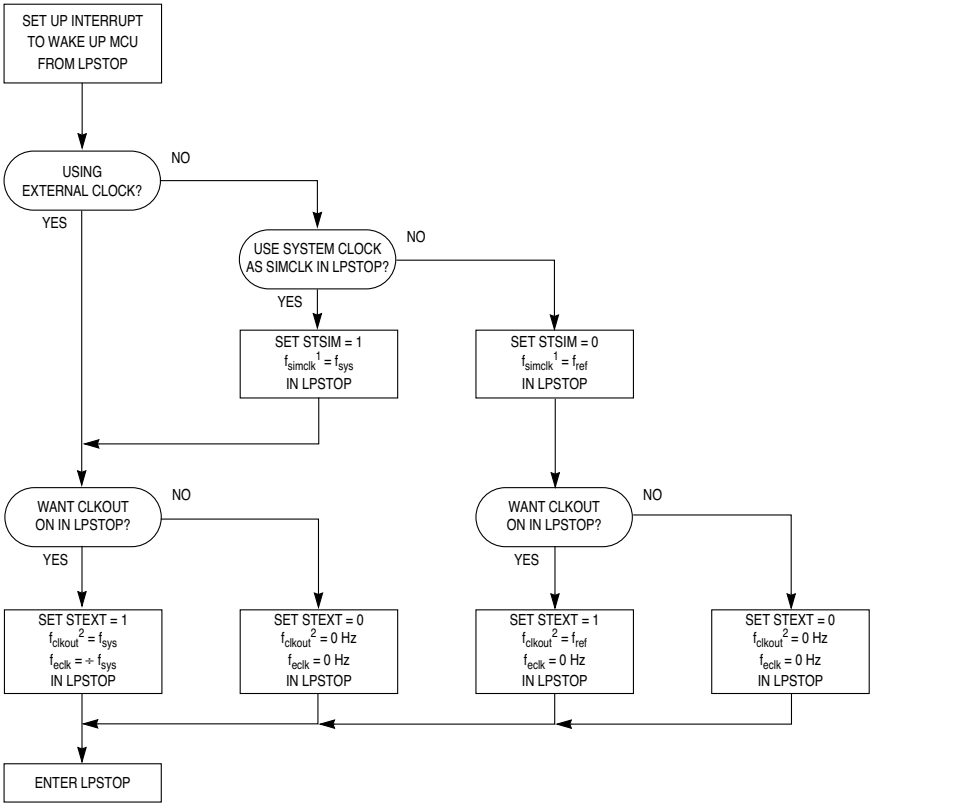
- External bus support
- Programmable chip select outputs
- System protection logic
- Watchdog timer, clock monitor and bus monitor
- Two 8-bit dual function input/output ports
- One 7-bit dual function output port
- Phase-locked loop (PLL) clock system

3.1.3 Standby RAM Module (SRAM)

- 4-Kbytes of static RAM
- No standby supply

3.1.4 Masked ROM Module (MRM)

- 8-Kbyte array, accessible as bytes or words
- User selectable default base address
- User selectable bootstrap ROM function
- User selectable ROM verification code



NOTES:
1. THE SIMCLK IS USED BY THE PIT, $\overline{IRQ}$, AND INPUT BLOCKS OF THE SIM.
2. CLKOUT CONTROL DURING LPSTOP IS OVERRIDDEN BY THE EXOFF BIT IN SIMCR. IF EXOFF = 1, THE CLKOUT PIN IS ALWAYS IN A HIGH IMPEDANCE STATE AND STEXT HAS NO EFFECT IN LPSTOP. IF EXOFF = 0, CLKOUT IS CONTROLLED BY STEXT IN LPSTOP.

LPSTOPFLOW

Figure 5-5 LPSTOP Flowchart

5.4.8 Low-Power STOP Mode Operation

When the CPU32 executes the LPSTOP instruction, the current interrupt priority mask is stored in the clock control logic, internal clocks are disabled according to the state of the STSIM bit in the SYNCR, and the MCU enters low-power stop mode. The bus monitor, halt monitor, and spurious interrupt monitor are all inactive during low-power stop mode.

During low-power stop mode, the clock input to the software watchdog timer is disabled and the timer stops. The software watchdog begins to run again on the first rising clock edge after low-power stop mode ends. The watchdog is not reset by low-power stop mode. A service sequence must be performed to reset the timer.

The periodic interrupt timer does not respond to the LPSTOP instruction, but continues to run during LPSTOP. To stop the periodic interrupt timer, PITR must be loaded with a zero value before the LPSTOP instruction is executed. A PIT interrupt, or an external interrupt request, can bring the MCU out of low-power stop mode if it has a higher priority than the interrupt mask value stored in the clock control logic when low-power stop mode is initiated. LPSTOP can be terminated by a reset.

5.5 External Bus Interface

The external bus interface (EBI) transfers information between the internal MCU bus and external devices. **Figure 5-8** shows a basic system with external memory and peripherals.

If bus termination signals remain unasserted, the MCU will continue to insert wait states, and the bus cycle will never end. If no peripheral responds to an access, or if an access is invalid, external logic should assert the BERR or HALT signals to abort the bus cycle (when BERR and HALT are asserted simultaneously, the CPU32 acts as though only BERR is asserted). When enabled, the SIM bus monitor asserts BERR when DSACK response time exceeds a predetermined limit. The bus monitor timeout period is determined by the BMT[1:0] field in SYPCR. The maximum bus monitor timeout period is 64 system clock cycles.

5.6.2.1 Read Cycle

During a read cycle, the MCU transfers data from an external memory or peripheral device. If the instruction specifies a long-word or word operation, the MCU attempts to read two bytes at once. For a byte operation, the MCU reads one byte. The portion of the data bus from which each byte is read depends on operand size, peripheral address, and peripheral port size. **Figure 5-10** is a flowchart of a word read cycle. Refer to **5.5.2 Dynamic Bus Sizing**, **5.5.4 Misaligned Operands**, and the *SIM Reference Manual* (SIMRM/AD) for more information.

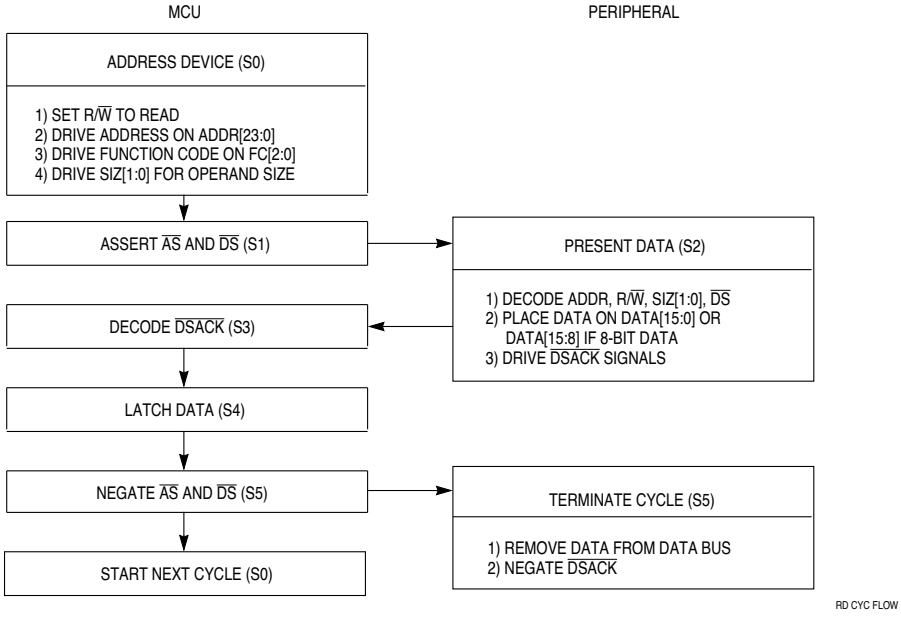


Figure 5-10 Word Read Cycle Flowchart

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.

To prepare the QADC for a scan sequence, the desired channel conversions are written to the CCW table. Software establishes the criteria for initiating the queue execution by programming queue operating mode. The queue operating mode determines what type of trigger event initiates queue execution.

A scan sequence may be initiated by the following trigger events:

- A software command
- Expiration of the periodic/interval timer
- An external trigger signal

Software also specifies whether the QADC is to perform a single pass through the queue or is to scan continuously. When a single-scan mode is selected, queue execution begins when software sets the single-scan enable bit. When a continuous-scan mode is selected, the queue remains active in the selected queue operating mode after the QADC completes each queue scan sequence.

During queue execution, the QADC reads each CCW from the active queue and executes conversions in four stages:

1. Initial sample
2. Transfer
3. Final sample
4. Resolution

During initial sample, 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 sample period, the sample capacitor and amplifier are bypassed, and the multiplexer input charges the RC DAC array directly. Each CCW specifies a final input sample time of 2, 4, 8, or 16 QCLK cycles. When an analog-to-digital conversion is complete, the result is written to the corresponding location in the result word table. The QADC continues to sequentially execute each CCW in the queue until the end of the queue is detected or a pause bit is found in a CCW.

When the pause bit is set in the current CCW, the QADC stops execution of the queue until a new trigger event occurs. The pause status flag bit is set, which may generate an interrupt request to notify software that the queue has reached the pause state. When the next trigger event occurs, the paused state ends, and the QADC continues to execute each CCW in the queue until another pause is encountered or the end of the queue is detected.

An end-of-queue condition is indicated as follows:

- The CCW channel field is programmed with 63 ($3F) to specify the end of the queue.
- The end of queue 1 is implied by the beginning of queue 2, which is specified in the BQ2 field in QACR2.
- The physical end of the queue RAM space defines the end of either queue.

10.7.7 MCSM Registers

The MCSM contains a status/interrupt/control register, a counter, and a modulus latch. All unused bits and reserved address locations return zero when read. Writes to unused bits and reserved address locations have no effect. The CTM4 contains three MCSMs, each with its own set of registers. Refer to **D.7.8 MCSM Status/Interrupt/Control Registers**, **D.7.9 MCSM Counter Registers**, and **D.7.10 MCSM Modulus Latch Registers** for information concerning MCSM register and bit descriptions.

10.8 Double-Action Submodule (DASM)

The double-action submodule (DASM) allows two 16-bit input capture or two 16-bit output compare functions to occur automatically without software intervention. The input edge detector can be programmed to trigger the capture function on user-specified edges. The output flip flop can be set by one of the output compare functions, and reset by the other one. Interrupt requests can optionally be generated by the input capture and the output compare functions. The user can select one of two incoming time bases for the input capture and output compare functions.

Six operating modes allow the DASM input capture and output compare functions to perform pulse width measurement, period measurement, single pulse generation, and continuous pulse width modulation, as well as standard input capture and output compare. The DASM can also function as a single I/O pin.

DASM operating mode is determined by the mode select field (MODE[3:0]) in the DASM status/interrupt/control register (DASMSIC). **Table 10-2** shows the different DASM operating modes.

Table 10-2 DASM Modes of Operation

MODE[3:0]	Mode	Description of Mode
0000	DIS	Disabled — Input pin is high impedance; IN gives state of input pin
0001	IPWM	Input pulse width measurement — Capture on leading edge and the trailing edge of an input pulse
0010	IPM	Input period measurement — Capture two consecutive rising/falling edges
0011	IC	Input capture — Capture when the designated edge is detected
0100	OCB	Output compare, flag set on B compare — Generate leading and trailing edges of an output pulse and set the flag
0101	OCAB	Output compare, flag set on A and B compare — Generate leading and trailing edges of an output pulse and set the flag
0110	—	Reserved
0111	—	Reserved
1xxx	OPWM	Output pulse width modulation — Generate continuous PWM output with 7, 9, 11, 12, 13, 14, 15, or 16 bits of resolution

The DASM is composed of two timing channels (A and B), an output flip-flop, an input edge detector, some control logic and an interrupt interface. All control and status bits are contained in DASMSIC.

Channel A consists of one 16-bit data register and one 16-bit comparator. To the user, channel B also appears to consist of one 16-bit data register and one 16-bit compar-

Refer to TPU programming note *Input Capture/Input Transition Counter (ITC) TPU Function* (TPUPN16/D) for more information.

11.4.3 Output Compare (OC)

The output compare function generates a rising edge, a falling edge, or a toggle of the previous edge in one of three ways:

1. Immediately upon CPU32 initiation, thereby generating a pulse with a length equal to a programmable delay time.
2. At a programmable delay time from a user-specified time.
3. As a continuous square wave. Upon receiving a link from a channel, OC references, without CPU32 interaction, a specifiable period and calculates an offset:

$$\text{OFFSET} = \text{PERIOD} \cdot \text{RATIO}$$

where “RATIO” is a parameter supplied by the user.

This algorithm generates a 50% duty-cycle continuous square wave with each high/low time equal to the calculated offset. Due to offset calculation, there is an initial link time before continuous pulse generation begins.

Refer to TPU programming note *Output Compare (OC) TPU Function* (TPUPN12/D) for more information.

11.4.4 Pulse-Width Modulation (PWM)

The TPU can generate a pulse-width modulated waveform with any duty cycle from zero to 100% (within the resolution and latency capability of the TPU). To define the PWM, the CPU32 provides one parameter that indicates the period and another parameter that indicates the high time. Updates to one or both of these parameters can direct the waveform change to take effect immediately, or coherently beginning at the next low-to-high transition of the pin.

Refer to TPU programming note *Pulse-Width Modulation (PWM) TPU Function* (TPUPN17/D) for more information.

11.4.5 Synchronized Pulse-Width Modulation (SPWM)

The TPU generates a PWM waveform in which the CPU32 can change the period and/or high time at any time. When synchronized to a time function on a second channel, the synchronized PWM low-to-high transitions have a time relationship to transitions on the second channel.

Refer to TPU programming note *Synchronized Pulse-Width Modulation (SPWM) TPU Function* (TPUPN19/D) for more information.

11.5.9 Frequency Measurement (FQM)

FQM counts the number of input pulses to a TPU channel during a user-defined window period. The function has single shot and continuous modes. No pulses are lost between sample windows in continuous mode. The user selects whether to detect pulses on the rising or falling edge. This function is intended for high speed measurement; measurement of slow pulses with noise rejection can be made with PTA.

Refer to TPU programming note *Frequency Measurement (FQM) TPU Function* (TPUPN03/D) for more information.

11.5.10 Hall Effect Decode (HALLD)

This function decodes the sensor signals from a brushless motor, along with a direction input from the CPU32, into a state number. The function supports two- or three-sensor decoding. The decoded state number is written into a COMM channel, which outputs the required commutation drive signals. In addition to brushless motor applications, the function can have more general applications, such as decoding option switches.

Refer to TPU programming note *Hall Effect Decode (HALLD) TPU Function* (TPUPN10/D) for more information.

11.6 Host Interface Registers

The TPU memory map contains three groups of registers:

- System configuration registers
- Channel control and status registers
- Development support and test verification registers

All registers except the channel interrupt status register (CISR) must be read or written by means of word accesses. The address space of the TPU memory map occupies 512 bytes. Unused registers within the 512-byte address space return zeros when read.

11.6.1 System Configuration Registers

The TPU configuration control registers, TPUMCR and TICR, determine the value of the prescaler, perform emulation control, specify whether the external TCR2 pin functions as a clock source or as gate of the DIV8 clock for TCR2, and determine interrupt request level and interrupt vector number assignment. Refer to **D.8.1 TPU Module Configuration Register** and **D.8.5 TPU Interrupt Configuration Register** for more information about TPUMCR and TICR.

11.6.1.1 Prescaler Control for TCR1

Timer count register one (TCR1) is clocked from the output of a prescaler. Two fields in TPUMCR control TCR1. The prescaler's input is the internal TPU system clock divided by either 4 or 32, depending on the value of the PSCK bit. The prescaler divides this input by 1, 2, 4, or 8, depending on the value of TCR1P. Channels using

TCR1 have the capability to resolve down to the TPU system clock divided by 4. Refer to **Figure 11-2** and **Table 11-1**.

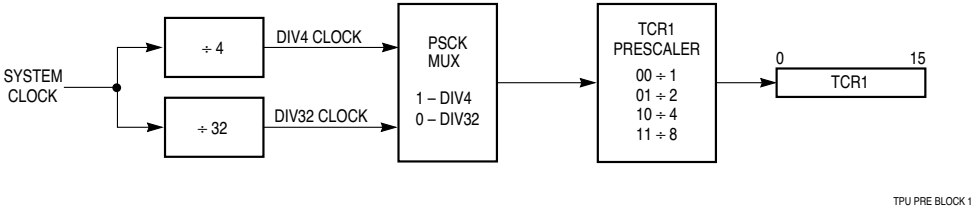


Figure 11-2 TCR1 Prescaler Control

Table 11-1 TCR1 Prescaler Control

		PSCK = 0		PSCK = 1	
TCR1 Prescaler	**Divide By**	**Number of Clocks**	**Rate at 20.97 MHz**	**Number of Clocks**	**Rate at 20.97 MHz**
00	1	32	1.6 µs	4	200 ns
01	2	64	3.2 µs	8	400 ns
10	4	128	6.4 µs	16	0.8 µs
11	8	256	12.8 µs	32	1.6 µs

11.6.1.2 Prescaler Control for TCR2

Timer count register two (TCR2), like TCR1, is clocked from the output of a prescaler. The T2CG bit in TPUMCR determines whether the T2CLK pin functions as an external clock source for TCR2 or as the gate in the use of TCR2 as a gated pulse accumulator. The function of the T2CG bit is shown in **Figure 11-3**.

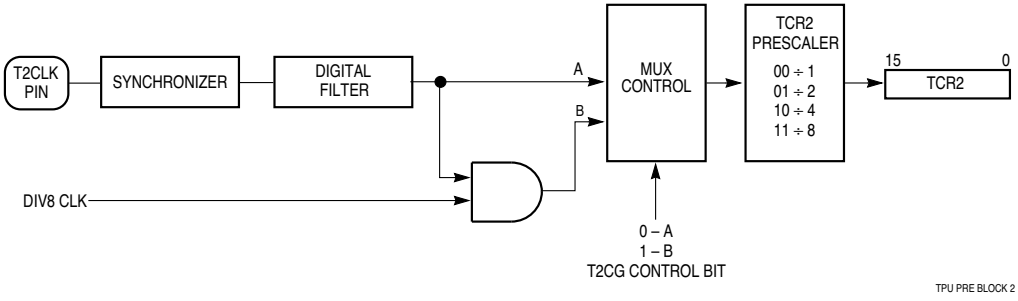


Figure 11-3 TCR2 Prescaler Control

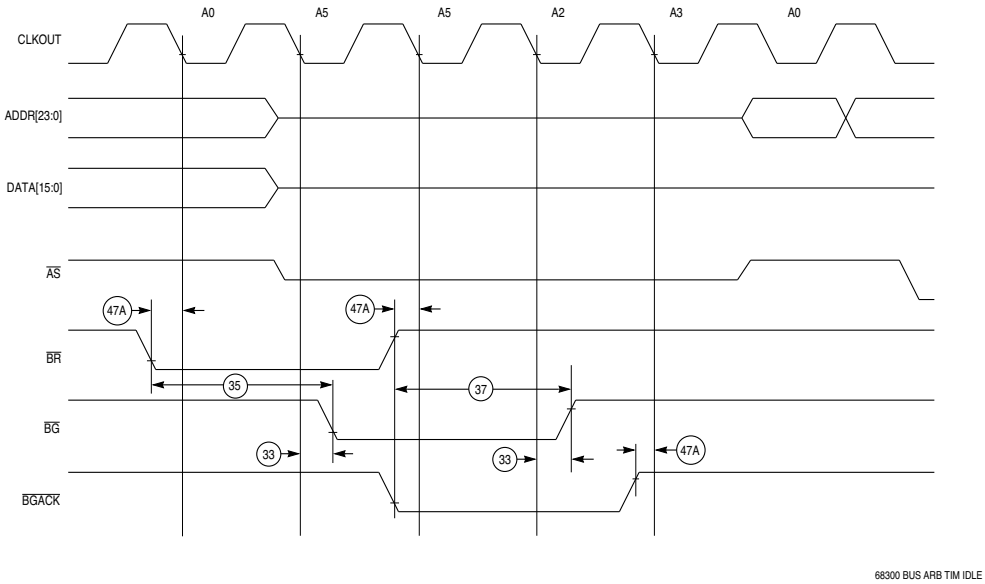


Figure A-9 Bus Arbitration Timing Diagram — Idle Bus Case

Table A-14 QADC Conversion Characteristics (Operating)

(V_{DDI} and V_{DDA} = 5.0 Vdc ± 5%, V_{SSI} and V_{SSA} = 0 Vdc, $T_A = T_L$ to T_H,
0.5 MHz ≤ f_{QCLK} ≤ 2.1 MHz, 2 clock input sample time)

Num	Parameter	Symbol	Min	Typ	Max	Unit
1	Resolution[1]	1 Count	—	5	—	mV
2	Differential nonlinearity[2]	DNL	—	—	± 0.5	Counts
3	Integral nonlinearity	INL	—	—	± 2.0	Counts
4	Absolute error[2, 3, 4] f_{QCLK} = 0.999 MHz[5] PQA PQB f_{QCLK} = 2.097 MHz[6] PQA PQB	AE	 — — — —	 — — — —	 ± 2.5 ± 2.5 ± 4.0 ± 4.0	Counts
5	Source impedance at input[7]	R_S	—	20	—	kΩ

NOTES:

1. At $V_{RH} - V_{RL}$ = 5.12 V, one count = 5 mV.
2. This parameter is periodically sampled rather than 100% tested.
3. Absolute error includes 1/2 count (2.5 mV) of inherent quantization error and circuit (differential, integral, and offset) error. Specification assumes that adequate low-pass filtering is present on analog input pins — capacitive filter with 0.01 µF to 0.1 µF capacitor between analog input and analog ground, typical source isolation impedance of 20 kΩ.
4. Assumes f_{sys} = 20.97 MHz.
5. Assumes clock prescaler values of:
   QACR0: PSH = %01111, PSA = %1, PSL = 100)
   CCW: BYP = %0
6. Assumes clock prescaler values of:
   QACR0: PSH = %00110, PSA = %1, PSL = 010)
   CCW: BYP = %0
7. Maximum source impedance is application-dependent. Error resulting from pin leakage depends on junction leakage into the pin and on leakage due to charge-sharing with internal capacitance.
   Error from junction leakage is a function of external source impedance and input leakage current. In the following expression, expected error in result value due to junction leakage is expressed in voltage (V_{errj}):
   $$V_{errj} = R_S \text{ X } I_{OFF}$$
   where I_{OFF} is a function of operating temperature. Refer to **Table A-12**.
   Charge-sharing leakage is a function of input source impedance, conversion rate, change in voltage between successive conversions, and the size of the decoupling capacitor used. Error levels are best determined empirically. In general, continuous conversion of the same channel may not be compatible with high source impedance.

Table A-17 SASM Timing Characteristics

(V_{DD} = 5.0 Vdc ± 5%, V_{SS} = 0Vdc, $T_A = T_L$ to T_H)

Num	Parameter	Symbol	Min	Max	Unit
1	Input pin low time	t_{PINL}	$2.0/f_{sys}$	—	µs
2	Input pin high time	t_{PINH}	$2.0/f_{sys}$	—	µs
3	Input capture resolution[1]	t_{RESCA}	—	$2.0/f_{sys}$	µs
4	Pin to input capture delay	t_{PCAPT}	$2.5/f_{sys}$	$4.5/f_{sys}$	µs
5	Pin to FLAG set	t_{PFLAG}	$2.5/f_{sys}$	$4.5/f_{sys}$	µs
6	Pin to IN bit delay	t_{PINB}	$1.5/f_{sys}$	$2.5/f_{sys}$	µs
7	OCT output pulse	t_{OCT}	$2.0/f_{sys}$	—	µs
8	Compare resolution	t_{RESCM}		$2.0/f_{sys}$	µs
9	TBB change to FLAG set	t_{CFLAG}	$1.5/f_{sys}$	$1.5/f_{sys}$	µs
10	TBB change to pin change[2]	t_{CPIN}	$1.5/f_{sys}$	$1.5/f_{sys}$	µs
11	FLAG to IMB interrupt request	t_{FIRQ}	$1.0/f_{sys}$	$1.0/f_{sys}$	µs

NOTES:
1. Minimum resolution depends on counter and prescaler divide ratio selection.
2. Time given from when new value is stable on time base bus.

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)	

QACR1 — Control Register 1 **$YFF20C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CIE1	PIE1	SSE1	NOT USED		MQ1[2:0]			RESERVED							
RESET:															
0	0	0			0	0	0								

CIE1 — Queue 1 Completion Interrupt Enable

CIE1 enables completion interrupts for queue 1. The interrupt request is generated when the conversion is complete for the last CCW in queue 1.

0 = Queue 1 completion interrupts disabled.

1 = Generate an interrupt request after completing the last CCW in queue 1.

PIE1 — Queue 1 Pause Interrupt Enable

PIE1 enables pause interrupts for queue 1. The interrupt request is generated when the conversion is complete for a CCW that has the pause bit set.

0 = Queue 1 pause interrupts disabled.

1 = Generate an interrupt request after completing a CCW in queue 1 which has the pause bit set.

SSE1 — Queue 1 Single-Scan Enable

SSE1 enables a single-scan of queue 1 after a trigger event occurs. The SSE1 bit may be set to a one during the same write cycle that sets the MQ1[2:0] bits for the single-scan queue operating mode. The single-scan enable bit can be written as a one or a zero, but is always read as a zero.

The SSE1 bit allows a trigger event to initiate queue execution for any single-scan operation on queue 1. The QADC clears SSE1 when the single-scan is complete.

MQ1[2:0] — Queue 1 Operating Mode

The MQ1 field selects the queue operating mode for queue 1. **Table D-25** shows the different queue 1 operating modes.

Table D-25 Queue 1 Operating Modes

MQ1[2:0]	Queue 1 Operating Mode
000	Disabled mode, conversions do not occur
001	Software triggered single-scan mode (started with SSE1)
010	External trigger rising edge single-scan mode (on ETRIG1 pin)
011	External trigger falling edge single-scan mode (on ETRIG1 pin)
100	Reserved mode, conversions do not occur
101	Software triggered continuous-scan mode (started with SSE1)
110	External trigger rising edge continuous-scan mode (on ETRIG1 pin)
111	External trigger falling edge continuous-scan mode (on ETRIG1 pin)

QACR2 — Control Register 2 **$YFF20E**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CIE2	PIE2	SSE2	MQ2[4:0]					RES	NOT USED	BQ2[5:0]					
RESET:															
0	0	0	0	0	0	0	0	0		1	0	0	1	1	1

CIE2 — Queue 2 Completion Interrupt Enable

CIE2 enables completion interrupts for queue 2. The interrupt request is generated when the conversion is complete for the last CCW in queue 2.

0 = Queue 2 completion interrupts disabled.

1 = Generate an interrupt request after completing the last CCW in queue 2.

PIE2 — Queue 2 Pause Interrupt Enable

PIE2 enables pause interrupts for queue 2. The interrupt request is generated when the conversion is complete for a CCW that has the pause bit set.

0 = Queue 2 pause interrupts disabled.

1 = Generate an interrupt request after completing a CCW in queue 2 which has the pause bit set.

SSE2 — Queue 2 Single-Scan Enable Bit

SSE2 enables a single-scan of queue 2 after a trigger event occurs. The SSE2 bit may be set to a one during the same write cycle that sets the MQ2[4:0] bits for the single-scan queue operating mode. The single-scan enable bit can be written as a one or a zero, but is always read as a zero.

The SSE2 bit allows a trigger event to initiate queue execution for any single-scan operation on queue 2. The QADC clears SSE2 when the single-scan is complete.

MQ2[4:0] — Queue 2 Operating Mode

The MQ2 field selects the queue operating mode for queue 2. **Table D-26** shows the bits in the MQ2 field which enable different queue 2 operating modes.

D.6.7 SCI Status Register

SCSR — SCI Status Register **$YFFC0C**

15 … 9	8	7	6	5	4	3	2	1	0
NOT USED	TDRE	TC	RDRF	RAF	IDLE	OR	NF	FE	PF
RESET:									
0 … 0	1	1	0	0	0	0	0	0	0

SCSR contains flags that show SCI operating conditions. These flags are cleared either by SCI hardware or by a read/write sequence. The sequence consists of reading SCSR, then reading or writing SCDR.

If an internal SCI signal for setting a status bit comes after reading the asserted status bits, but before writing or reading SCDR, the newly set status bit is not cleared. SCSR must be read again with the bit set and SCDR must be read or written before the status bit is cleared.

A long-word read can consecutively access both SCSR and SCDR. This action clears receive status flag bits that were set at the time of the read, but does not clear TDRE or TC flags. Reading either byte of SCSR causes all 16 bits to be accessed, and any status bit already set in either byte is cleared on a subsequent read or write of SCDR.

TDRE — Transmit Data Register Empty
- 0 = Transmit data register still contains data to be sent to the transmit serial shifter.
- 1 = A new character can now be written to the transmit data register.

TC — Transmit Complete
- 0 = SCI transmitter is busy.
- 1 = SCI transmitter is idle.

RDRF — Receive Data Register Full
- 0 = Receive data register is empty or contains previously read data.
- 1 = Receive data register contains new data.

RAF — Receiver Active
- 0 = SCI receiver is idle.
- 1 = SCI receiver is busy.

IDLE — Idle-Line Detected
- 0 = SCI receiver did not detect an idle-line condition.
- 1 = SCI receiver detected an idle-line condition.

OR — Overrun Error
- 0 = Receive data register is empty and can accept data from the receive serial shifter.
- 1 = Receive data register is full and cannot accept data from the receive serial shifter. Any data in the shifter is lost and RDRF remains set.

NF — Noise Error Flag
- 0 = No noise detected in the received data.
- 1 = Noise detected in the received data.

DIV23 — Divide By 2/Divide By 3

The DIV23 bit is a read/write control bit that selects the division ratio of the first prescaler stage. It may be changed at any time.

0 = First prescaler stage divides by two.

1 = First prescaler stage divides by three.

PSEL[1:0] — Prescaler Division Ratio Select

This bit field selects the division ratio of the programmable prescaler output signal PCLK6. Refer to **Table D-38**.

Table D-38 Prescaler Division Ratio Select Field

Prescaler Control Bits			Prescaler Division Ratio					
DIV23	**PSEL1**	**PSEL0**	**PCLK1**	**PCLK2**	**PCLK3**	**PCLK4**	**PCLK5**	**PCLK6**
0	0	0	2	4	8	16	32	64
0	0	1	2	4	8	16	32	128
0	1	0	2	4	8	16	32	256
0	1	1	2	4	8	16	32	512
1	0	0	3	6	12	24	48	96
1	0	1	3	6	12	24	48	192
1	1	0	3	6	12	24	48	384
1	1	1	3	6	12	24	48	768

D.7.5 CPSM Test Register

CPTR — CPSM Test Register **$YFF40A**

Used only during factory test.

D.7.6 FCSM Status/Interrupt/Control Register

FCSMSIC — FCSM Status/Interrupt/Control Register **$YFF460**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COF	IL[2:0]			IARB3	NOT USED	DRVA	DRVB	IN	NOT USED				CLK[2:0]		
RESET:															
U	0	0	0	0		0	0	U					0	0	0

COF — Counter Overflow Flag

This flag indicates whether or not a counter overflow has occurred. An overflow is defined as the transition of the counter from $FFFF to $0000. If the IL[2:0] field is nonzero, an interrupt request is generated when the COF bit is set.

0 = Counter overflow has not occurred

1 = Counter overflow has occurred

This flag bit is set only by hardware and cleared by software or system reset. To clear the flag, first read the bit as a one, then write a zero to the bit.

D.8 Time Processor Unit (TPU)

Table D-51 shows the TPU 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-51 TPU Register Map

Access	Address[1]	15 0
S	$YFFE00	Module Configuration Register (TPUMCR)
S	$YFFE02	Test Configuration Register (TCR)
S	$YFFE04	Development Support Control Register (DSCR)
S	$YFFE06	Development Support Status Register (DSSR)
S	$YFFE08	TPU Interrupt Configuration Register (TICR)
S	$YFFE0A	Channel Interrupt Enable Register (CIER)
S	$YFFE0C	Channel Function Selection Register 0 (CFSR0)
S	$YFFE0E	Channel Function Selection Register 1 (CFSR1)
S	$YFFE10	Channel Function Selection Register 2 (CFSR2)
S	$YFFE12	Channel Function Selection Register 3 (CFSR3)
S/U	$YFFE14	Host Sequence Register 0 (HSQR0)
S/U	$YFFE16	Host Sequence Register 1 (HSQR1)
S/U	$YFFE18	Host Service Request Register 0 (HSRR0)
S/U	$YFFE1A	Host Service Request Register 1 (HSRR1)
S	$YFFE1C	Channel Priority Register 0 (CPR0)
S	$YFFE1E	Channel Priority Register 1 (CPR1)
S	$YFFE20	Channel Interrupt Status Register (CISR)
S	$YFFE22	Link Register (LR)
S	$YFFE24	Service Grant Latch Register (SGLR)
S	$YFFE26	Decoded Channel Number Register (DCNR)

NOTES:
1. Y = M111, where M represents the logic state of the module mapping (MM) bit in the SIMCR.

D.8.1 TPU Module Configuration Register

TPUMCR — TPU Module Configuration Register **$YFFE00**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOP	TCR1P[1:0]		TCR2P[1:0]		EMU	T2CG	STF	SUPV	PSCK	0	0	IARB[3:0]			
RESET:															
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

STOP — Low-Power Stop Mode Enable
0 = Enable TPU clocks.
1 = Disable TPU clocks.

–N–

–O–

–P–