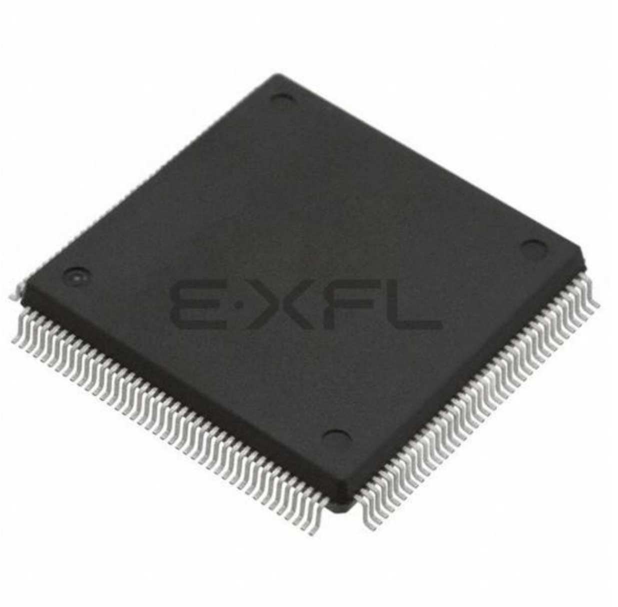
NXP USA Inc. - SC68376BACAB20 Datasheet



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	Not For New Designs
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/sc68376bacab20

Paragraph Title Page

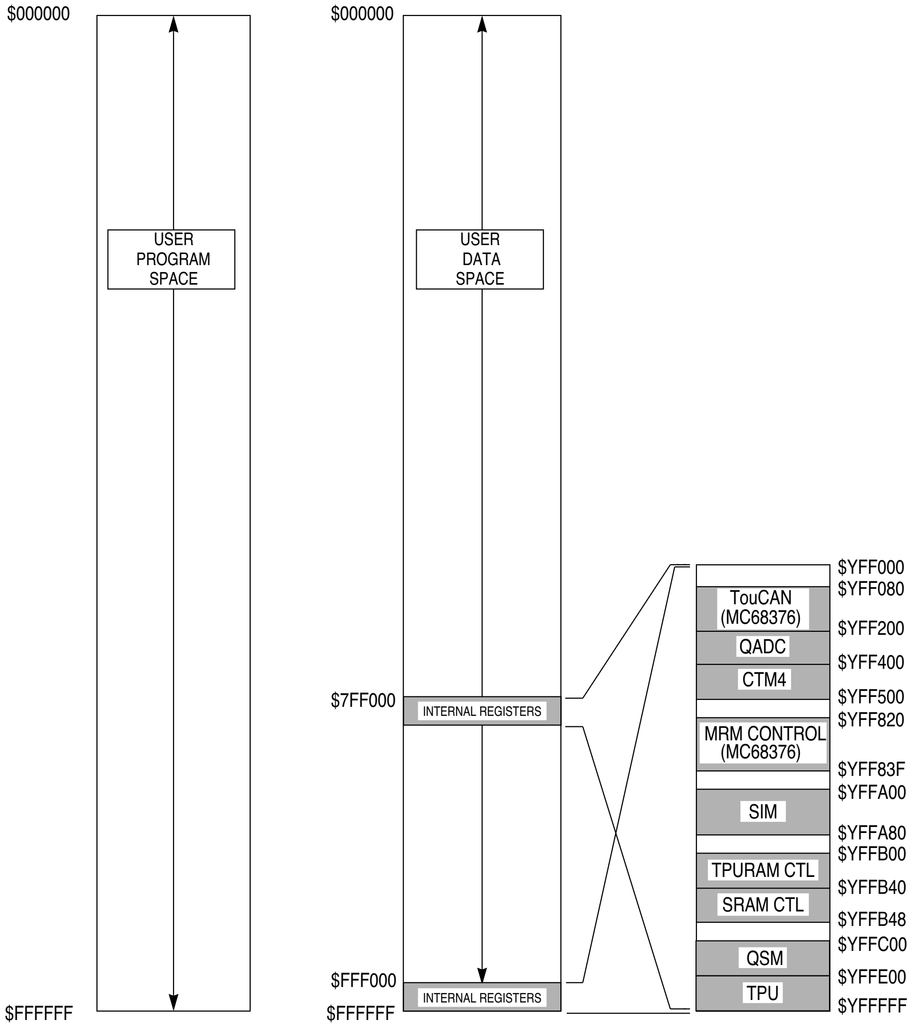
SECTION 6 STANDBY RAM MODULE

SECTION 7 MASKED ROM MODULE

SECTION 8 QUEUED ANALOG-TO-DIGITAL CONVERTER MODULE

LIST OF TABLES

Table Title Page



NOTES:

1. LOCATION OF THE MODULE CONTROL REGISTERS IS DETERMINED BY THE STATE OF THE MODULE MAPPING (MM) BIT IN THE SIM CONFIGURATION REGISTER. Y = M111, WHERE M IS THE STATE OF THE MM BIT.
2. UNUSED ADDRESSES WITHIN THE INTERNAL REGISTER BLOCK ARE MAPPED EXTERNALLY. “RESERVED” BLOCKS ARE NOT MAPPED EXTERNALLY.
3. SOME INTERNAL REGISTERS ARE NOT AVAILABLE IN USER SPACE.

336/376 USER P/D MAP

Figure 3-8 User Space (Separate Program/Data Space) Map

4.4 Virtual Memory

The full addressing range of the CPU32 on the MC68336/376 is 16 Mbytes in each of eight address spaces. Even though most systems implement a smaller physical memory, the system can be made to appear to have a full 16 Mbytes of memory available to each user program by using virtual memory techniques.

A system that supports virtual memory has a limited amount of high-speed physical memory that can be accessed directly by the processor and maintains an image of a much larger virtual memory on a secondary storage device. When the processor attempts to access a location in the virtual memory map that is not resident in physical memory, a page fault occurs. The access to that location is temporarily suspended while the necessary data is fetched from secondary storage and placed in physical memory. The suspended access is then restarted or continued.

The CPU32 uses instruction restart, which requires that only a small portion of the internal machine state be saved. After correcting the fault, the machine state is restored, and the instruction is fetched and started again. This process is completely transparent to the application program.

4.5 Addressing Modes

Addressing in the CPU32 is register-oriented. Most instructions allow the results of the specified operation to be placed either in a register or directly in memory. There is no need for extra instructions to store register contents in memory.

There are seven basic addressing modes:

- Register Direct
- Register Indirect
- Register Indirect with Index
- Program Counter Indirect with Displacement
- Program Counter Indirect with Index
- Absolute
- Immediate

The register indirect addressing modes include postincrement, predecrement, and offset capability. The program counter indirect mode also has index and offset capabilities. In addition to these addressing modes, many instructions implicitly specify the use of the status register, stack pointer, and/or program counter.

4.6 Processing States

The processor is always in one of four processing states: normal, exception, halted, or background. The normal processing state is associated with instruction execution; the bus is used to fetch instructions and operands and to store results.

This additional $\overline{\text{BG}}$ assertion allows external arbitration circuitry to select the next bus master before the current master has released the bus.

Refer to **Figure 5-15**, which shows bus arbitration for a single device. The flowchart shows $\overline{\text{BR}}$ negated at the same time $\overline{\text{BGACK}}$ is asserted.

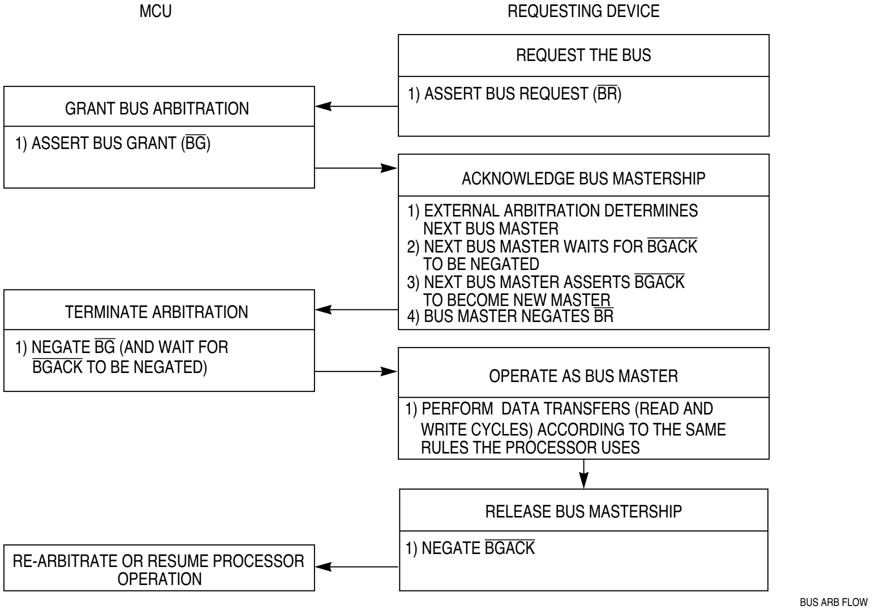

MCU
REQUESTING DEVICE
REQUEST THE BUS
1) ASSERT BUS REQUEST ($\overline{\text{BR}}$)
GRANT BUS ARBITRATION
1) ASSERT BUS GRANT ($\overline{\text{BG}}$)
ACKNOWLEDGE BUS MASTERSHIP
1) EXTERNAL ARBITRATION DETERMINES NEXT BUS MASTER
2) NEXT BUS MASTER WAITS FOR $\overline{\text{BGACK}}$ TO BE NEGATED
3) NEXT BUS MASTER ASSERTS $\overline{\text{BGACK}}$ TO BECOME NEW MASTER
4) BUS MASTER NEGATES $\overline{\text{BR}}$
TERMINATE ARBITRATION
1) NEGATE $\overline{\text{BG}}$ (AND WAIT FOR $\overline{\text{BGACK}}$ TO BE NEGATED)
OPERATE AS BUS MASTER
1) PERFORM DATA TRANSFERS (READ AND WRITE CYCLES) ACCORDING TO THE SAME RULES THE PROCESSOR USES
RELEASE BUS MASTERSHIP
1) NEGATE $\overline{\text{BGACK}}$
RE-ARBITRATE OR RESUME PROCESSOR OPERATION
BUS ARB FLOW

Figure 5-15 Bus Arbitration Flowchart for Single Request

5.6.6.1 Show Cycles

The MCU normally performs internal data transfers without affecting the external bus, but it is possible to show these transfers during debugging. $\overline{\text{AS}}$ is not asserted externally during show cycles.

Show cycles are controlled by SHEN[1:0] in SIMCR. This field is set to %00 by reset. When show cycles are disabled, the address bus, function codes, size, and read/write signals reflect internal bus activity, but $\overline{\text{AS}}$ and $\overline{\text{DS}}$ are not asserted externally and external data bus pins are in high-impedance state during internal accesses. Refer to **5.2.3 Show Internal Cycles** and the *SIM Reference Manual* (SIMRM/AD) for more information.

When show cycles are enabled, $\overline{\text{DS}}$ is asserted externally during internal cycles, and internal data is driven out on the external data bus. Because internal cycles normally continue to run when the external bus is granted, one SHEN encoding halts internal bus activity while there is an external master.

E. After arbitration, the interrupt acknowledge cycle is completed in one of the following ways:
   1. When there is no contention (IARB = %0000), the spurious interrupt monitor asserts $\overline{\text{BERR}}$, and the CPU32 generates the spurious interrupt vector number.
   2. The dominant interrupt source (external or internal) supplies a vector number and $\overline{\text{DSACK}}$ signals appropriate to the access. The CPU32 acquires the vector number.
   3. The $\overline{\text{AVEC}}$ signal is asserted (the signal can be asserted by the dominant external interrupt source or the pin can be tied low), and the CPU32 generates an autovector number corresponding to interrupt priority.
   4. The bus monitor asserts $\overline{\text{BERR}}$ and the CPU32 generates the spurious interrupt vector number.

F. The vector number is converted to a vector address.

G. The content of the vector address is loaded into the PC and the processor transfers control to the exception handler routine.

5.8.5 Interrupt Acknowledge Bus Cycles

Interrupt acknowledge bus cycles are CPU32 space cycles that are generated during exception processing. For further information about the types of interrupt acknowledge bus cycles determined by $\overline{\text{AVEC}}$ or $\overline{\text{DSACK}}$, refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** and the *SIM Reference Manual* (SIMRM/AD).

5.9 Chip-Selects

Typical microcontrollers require additional hardware to provide external chip-select and address decode signals. The MCU includes 12 programmable chip-select circuits that can provide 2 to 16 clock-cycle access to external memory and peripherals. Address block sizes of two Kbytes to one Mbyte can be selected. **Figure 5-19** is a diagram of a basic system that uses chip-selects.

tion to restart at the designated location. Reads of SPCR2 return the current value of the register, not of the buffer. Writing the same value into any control register except SPCR2 while the QSPI is enabled has no effect on QSPI operation.

9.3.1.2 Status Register

SPSR contains information concerning the current serial transmission. Only the QSPI can set the bits in this register. The CPU32 reads SPSR to obtain QSPI status information and writes SPSR to clear status flags.

9.3.2 QSPI RAM

The QSPI contains an 80-byte block of dual-port access static RAM that can be accessed by both the QSPI and the CPU32. The RAM is divided into three segments: receive data RAM, transmit data RAM, and command data RAM. Receive data is information received from a serial device external to the MCU. Transmit data is information stored for transmission to an external device. Command control data defines transfer parameters. Refer to **Figure 9-3**, which shows RAM organization.

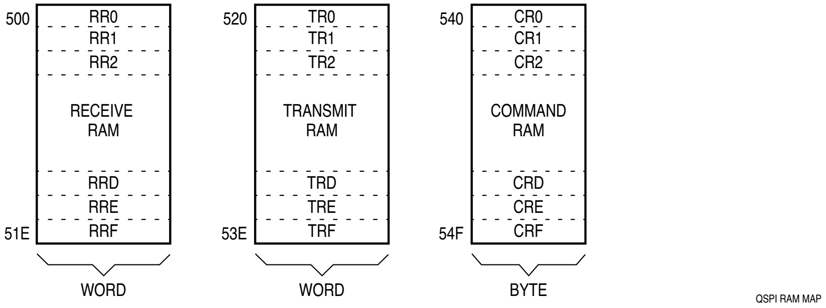


Figure 9-3 QSPI RAM

9.3.2.1 Receive RAM

Data received by the QSPI is stored in this segment. The CPU32 reads this segment to retrieve data from the QSPI. Data stored in the receive RAM is right-justified. Unused bits in a receive queue entry are set to zero by the QSPI upon completion of the individual queue entry. The CPU32 can access the data using byte, word, or long-word addressing.

The CPTQP value in SPSR shows which queue entries have been executed. The CPU32 uses this information to determine which locations in receive RAM contain valid data before reading them.

9.3.2.2 Transmit RAM

Data that is to be transmitted by the QSPI is stored in this segment and must be written to transmit RAM in a right-justified format. The QSPI cannot modify information in the transmit RAM. The QSPI copies the information to its data serializer for transmission. Information remains in transmit RAM until overwritten.

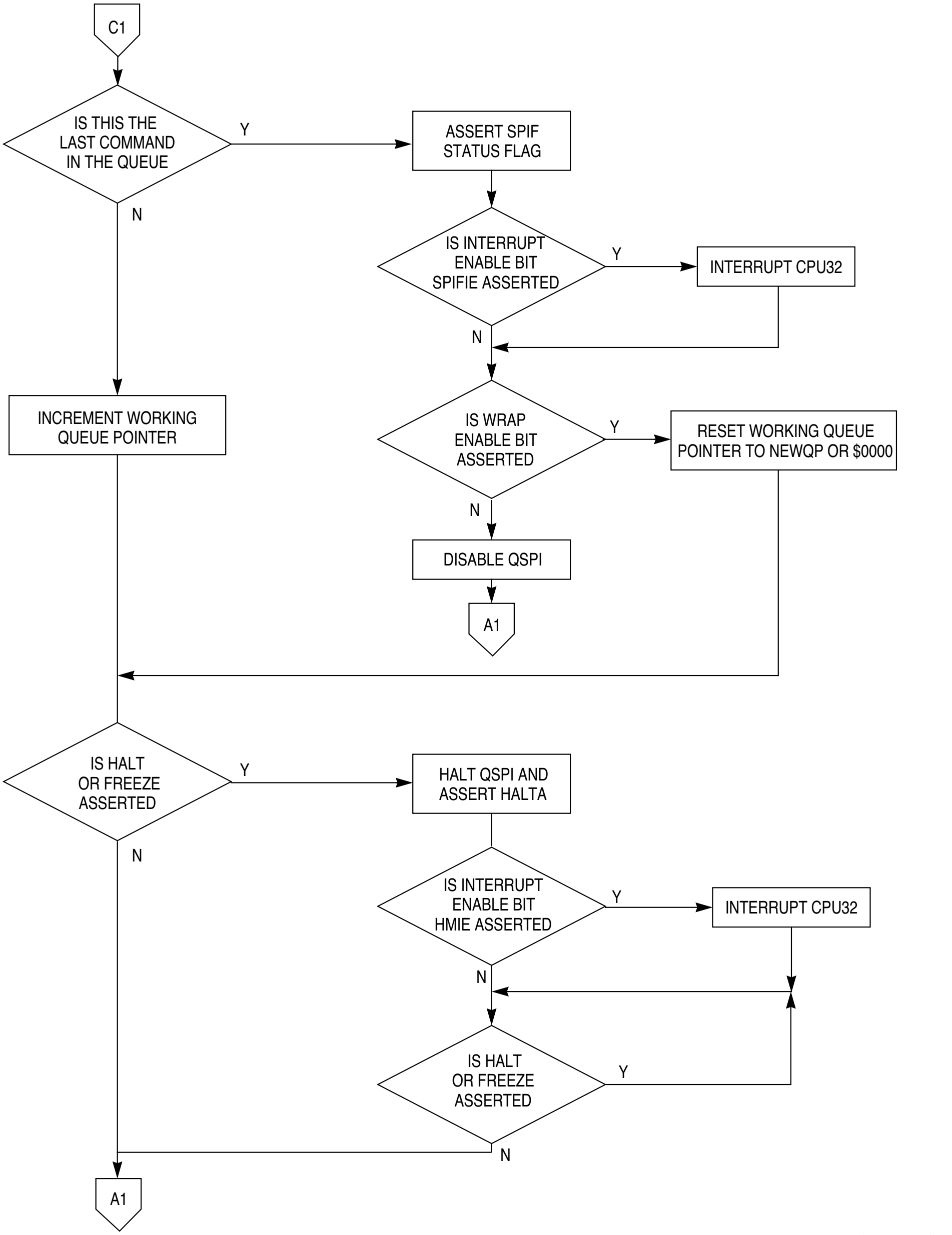

C1
IS THIS THE LAST COMMAND IN THE QUEUE
Y
N
ASSERT SPIF STATUS FLAG
IS INTERRUPT ENABLE BIT SPIFIE ASSERTED
Y
N
INTERRUPT CPU32
INCREMENT WORKING QUEUE POINTER
IS WRAP ENABLE BIT ASSERTED
Y
N
RESET WORKING QUEUE POINTER TO NEWQP OR $0000
DISABLE QSPI
A1
IS HALT OR FREEZE ASSERTED
Y
N
HALT QSPI AND ASSERT HALTA
IS INTERRUPT ENABLE BIT HMIE ASSERTED
Y
N
INTERRUPT CPU32
IS HALT OR FREEZE ASSERTED
Y
N
A1
QSPI MSTR3 FLOW4

Figure 9-7 Flowchart of QSPI Master Operation (Part 3)

9.3.5.2 Master Wrap-Around Mode

Wrap-around mode is enabled by setting the WREN bit in SPCR2. The queue can wrap to pointer address $0 or to the address pointed to by NEWQP, depending on the state of the WRTO bit in SPCR2.

In wrap-around mode, the QSPI cycles through the queue continuously, even while the QSPI is requesting interrupt service. SPE is not cleared when the last command in the queue is executed. New receive data overwrites previously received data in receive RAM. Each time the end of the queue is reached, the SPIF flag is set. SPIF is not automatically reset. If interrupt-driven QSPI service is used, the service routine must clear the SPIF bit to end the current interrupt request. Additional interrupt requests during servicing can be prevented by clearing SPIFIE, but SPIFIE is buffered. Clearing it does not end the current request.

Wrap-around mode is exited by clearing the WREN bit or by setting the HALT bit in SPCR3. Exiting wrap-around mode by clearing SPE is not recommended, as clearing SPE may abort a serial transfer in progress. The QSPI sets SPIF, clears SPE, and stops the first time it reaches the end of the queue after WREN is cleared. After HALT is set, the QSPI finishes the current transfer, then stops executing commands. After the QSPI stops, SPE can be cleared.

9.3.5.3 Slave Mode

Clearing the MSTR bit in SPCR0 selects slave mode operation. In slave mode, the QSPI is unable to initiate serial transfers. Transfers are initiated by an external SPI bus master. Slave mode is typically used on a multi-master SPI bus. Only one device can be bus master (operate in master mode) at any given time.

Before QSPI operation is initiated, QSM register PQSPAR must be written to assign necessary pins to the QSPI. The pins necessary for slave mode operation are MISO and MOSI, SCK, and PCS0/$\overline{\text{SS}}$. MISO is used for serial data output in slave mode, and MOSI is used for serial data input. Either or both may be necessary, depending on the particular application. SCK is the serial clock input in slave mode. Assertion of the active-low slave select signal $\overline{\text{SS}}$ initiates slave mode operation.

Before slave mode operation is initiated, DDRQS must be written to direct data flow on the QSPI pins used. Configure the MOSI, SCK and PCS0/$\overline{\text{SS}}$ pins as inputs. The MISO pin must be configured as an output.

After pins are assigned and configured, write data to be transmitted into transmit RAM. Command RAM is not used in slave mode, and does not need to be initialized. Set the queue pointers, as appropriate.

When SPE is set and MSTR is clear, a low state on the slave select PCS0/$\overline{\text{SS}}$ pin begins slave mode operation at the address indicated by NEWQP. Data that is received is stored at the pointer address in receive RAM. Data is simultaneously loaded into the data serializer from the pointer address in transmit RAM and transmitted. Transfer is synchronized with the externally generated SCK. The CPHA and CPOL bits determine upon which SCK edge to latch incoming data from the MISO pin and to drive outgoing data from the MOSI pin.

data space accesses. The SUPV bit in QADCMCR designates the assignable space as supervisor or unrestricted.

Attempts to read supervisor-only data space when the CPU32 is not in supervisor mode causes a value of $0000 to be returned. Attempts to read assignable data space when the CPU32 is not in supervisor mode and when the space is programmed as supervisor space, causes a value of $FFFF to be returned. Attempts to write supervisor-only or supervisor-assigned data space when the CPU32 is in user mode has no effect.

The supervisor-only data space segment contains the QADC global registers, which include QADCMCR, QADCTEST, and QADCINT. The supervisor/unrestricted space designation for the CCW table, the result word table, and the remaining QADC registers is programmable. Refer to **D.5.1 QADC Module Configuration Register** for more information.

8.6.4 Interrupt Arbitration Priority

Each module that can request interrupts, including the QADC, has an interrupt arbitration number (IARB) field in its module configuration register. Each IARB field must have a different non-zero value. During an interrupt acknowledge cycle, IARB permits arbitration among simultaneous interrupts of the same priority level.

The reset value of IARB in the QADCMCR is $0. Initialization software must set the IARB field to a non-zero value in order for QADC interrupts to be arbitrated. Refer to **D.5.1 QADC Module Configuration Register** for more information.

8.7 Test Register

The QADC test register (QADCTEST) is used only during factory testing of the MCU.

8.8 General-Purpose I/O Port Operation

QADC port pins, when used as general-purpose input, are conditioned by a synchronizer with an enable feature. The synchronizer is not enabled until the QADC decodes an IMB bus cycle which addresses the port data register to minimize the high-current effect of mid-level signals on the inputs used for analog signals. Digital input signals must meet the input low voltage (V_{IL}) or input high voltage (V_{IH}) specifications in **APPENDIX A ELECTRICAL CHARACTERISTICS**. If an analog input pin does not meet the digital input pin specifications when a digital port read operation occurs, an indeterminate state is read.

During a port data register read, the actual value of the pin is reported when its corresponding bit in the data direction register defines the pin to be an input (port A only). When the data direction bit specifies the pin to be an output, the content of the port data register is read. By reading the latch which drives the output pin, software instructions that read data, modify it, and write the result, like bit manipulation instructions, work correctly.

effect at the same time. Parameter RAM hardware supports coherent access of two adjacent 16-bit parameters. The host CPU must use a long-word operation to guarantee coherency.

11.3.6 Emulation Support

Although factory-programmed time functions can perform a wide variety of control tasks, they may not be ideal for all applications. The TPU provides emulation capability that allows the user to develop new time functions. Emulation mode is entered by setting the EMU bit in TPUMCR. In emulation mode, an auxiliary bus connection is made between TPURAM and the TPU, and access to TPURAM via the intermodule bus is disabled. A 9-bit address bus, a 32-bit data bus, and control lines transfer information between the modules. To ensure exact emulation, RAM module access timing remains consistent with access timing of the TPU microcode ROM control store.

To support changing TPU application requirements, Motorola has established a TPU function library. The function library is a collection of TPU functions written for easy assembly in combination with each other or with custom functions. Refer to Motorola Programming Note TPUPN00/D, *Using the TPU Function Library and TPU Emulation Mode* for information about developing custom functions and accessing the TPU function library. Refer to the *TPU Reference Manual* (TPURM/AD) and the Motorola TPU Literature Package (TPULITPAK/D) for more information about specific functions.

11.3.7 TPU Interrupts

Each of the TPU channels can generate an interrupt service request. Interrupts for each channel must be enabled by writing to the appropriate control bit in the channel interrupt enable register (CIER). The channel interrupt status register (CISR) contains one interrupt status flag per channel. Time functions set the flags. Setting a flag bit causes the TPU to make an interrupt service request if the corresponding channel interrupt enable bit is set and the interrupt request level is non-zero.

The value of the channel interrupt request level (CIRL) field in the TPU interrupt configuration register (TICR) determines the priority of all TPU interrupt service requests. CIRL values correspond to MCU interrupt request signals $\overline{\text{IRQ[7:1]}}$. $\overline{\text{IRQ7}}$ is the highest-priority request signal; $\overline{\text{IRQ1}}$ has the lowest priority. Assigning a value of %111 to CIRL causes $\overline{\text{IRQ7}}$ to be asserted when a TPU interrupt request is made; lower field values cause corresponding lower-priority interrupt request signals to be asserted. Assigning CIRL a value of %000 disables all interrupts.

The CPU32 recognizes only interrupt requests of a priority greater than the value contained in the interrupt priority (IP) mask in the status register. When the CPU32 acknowledges an interrupt request, the priority of the acknowledged interrupt is written to the IP mask and is driven out onto the IMB address lines.

When the IP mask value driven out on the address lines is the same as the CIRL value, the TPU contends for arbitration priority. The IARB field in TPUMCR contains the TPU arbitration number. Each module that can make an interrupt service request must be assigned a unique non-zero IARB value in order to implement an arbitration scheme.

Table A-9 QSPI Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H 200 pF load on all QSPI pins)[1]

Num	Function	Symbol	Min	Max	Unit
1	Operating Frequency Master Slave	f_{QSPI}	DC DC	1/4 1/4	f_{sys} f_{sys}
2	Cycle Time Master Slave	t_{qcyc}	4 4	510 —	t_{cyc} t_{cyc}
3	Enable Lead Time Master Slave	t_{lead}	2 2	128 —	t_{cyc} t_{cyc}
4	Enable Lag Time Master Slave	t_{lag}	— 2	1/2 —	SCK t_{cyc}
5	Clock (SCK) High or Low Time Master Slave[2]	t_{sw}	2 t_{cyc} – 60 2 t_{cyc} – n	255 t_{cyc} —	ns ns
6	Sequential Transfer Delay Master Slave (Does Not Require Deselect)	t_{td}	17 13	8192 —	t_{cyc} t_{cyc}
7	Data Setup Time (Inputs) Master Slave	t_{su}	30 20	— —	ns ns
8	Data Hold Time (Inputs) Master Slave	t_{hi}	0 20	— —	ns ns
9	Slave Access Time	t_a	—	1	t_{cyc}
10	Slave MISO Disable Time	t_{dis}	—	2	t_{cyc}
11	Data Valid (after SCK Edge) Master Slave	t_v	— —	50 50	ns ns
12	Data Hold Time (Outputs) Master Slave	t_{ho}	0 0	— —	ns ns
13	Rise Time Input Output	t_{ri} t_{ro}	— —	2 30	µs ns
14	Fall Time Input Output	t_{fi} t_{fo}	— —	2 30	µs ns

NOTES:

1. All AC timing is shown with respect to 20% V_{DD} and 70% V_{DD} levels unless otherwise noted.
2. For high time, n = External SCK rise time; for low time, n = External SCK fall time.

Table B-2 MC68376 Ordering Information

Part Number	Package Type	Frequency (MHz)	TPU	Mask ROM	Temperature	Package Order Quantity	Order Number
MC68376	160-pin QFP	20.97 MHz	A	Blank	–40 to +85 °C	2	SPMC68376BACFT20
						24	MC68376BACFT20
						120	MC68376BACFT20B1
					–40 to +105 °C	2	SPMC68376BAVFT20
						24	MC68376BAVFT20
						120	MC68376BAVFT20B1
					–40 to +125 °C	2	SPMC68376BAMFT20
						24	MC68376BAMFT20
						120	MC68376BAMFT20B1
			G	Blank	–40 to +85 °C	2	SPMC68376BGCFT20
						24	MC68376BGCFT20
						120	MC68376BGCFT20B1
					–40 to +105 °C	2	SPMC68376BGVFT20
						24	MC68376BGVFT20
						120	MC68376BGVFT20B1
					–40 to +125 °C	2	SPMC68376BGMFT20
						24	MC68376BGMFT20
						120	MC68376BGMFT20B1

D.2.3 Clock Synthesizer Control Register

SYNCR — Clock Synthesizer Control Register **$YFFA04**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W	X	Y[5:0]						EDIV	0	0	RSVD[1]	SLOCK	RSVD[1]	STSIM	STEXT
RESET:															
0	0	1	1	1	1	1	1	0	0	0	0	U	0	0	0

NOTES:

[1] Ensure that initialization software does not change the value of these bits. They should always be zero.

SYNCR determines system clock operating frequency and operation during low-power stop mode. Clock frequency is determined by SYNCR bit settings as follows:

$$f_{sys} = \frac{f_{ref}}{128}[4(Y+1)(2^{(2W+X)})]$$

W — Frequency Control (VCO)

This bit controls a prescaler tap in the synthesizer feedback loop. Setting this bit increases the VCO speed by a factor of four. VCO relock delay is required.

X — Frequency Control (Prescaler)

This bit controls a divide by two prescaler that is not in the synthesizer feedback loop. Setting the bit doubles clock speed without changing the VCO speed. No VCO relock delay is required.

Y[5:0] — Frequency Control (Counter)

The Y field controls the modulus down counter in the synthesizer feedback loop, causing it to divide by a value of Y + 1. VCO relock delay is required.

EDIV — E Clock Divide Rate

0 = ECLK frequency is system clock divided by 8.
1 = ECLK frequency is system clock divided by 16.

ECLK is an external M6800 bus clock available on ADDR23.

SLOCK — Synthesizer Lock Flag

0 = VCO is enabled, but has not locked.
1 = VCO has locked on the desired frequency or VCO is disabled.

The MCU remains in reset until the synthesizer locks, but SLOCK does not indicate synthesizer lock status until after the user writes to SYNCR.

STSIM — Stop Mode SIM Clock

0 = When LPSTOP is executed, the SIM clock is driven from the crystal oscillator and the VCO is turned off to conserve power.
1 = When LPSTOP is executed, the SIM clock is driven from the VCO.

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

D.9 Standby RAM Module with TPU Emulation Capability (TPURAM)

Table D-58 is the TPURAM address map. TPURAM responds to both program and data space accesses. The RASP bit in TRAMMCR determines whether the processor must be operating in supervisor mode to access the array. TPURAM control registers are accessible in supervisor mode only.

Table D-58 TPURAM Address Map

Address[1]	**15** **0**
$YFFB00	TPURAM Module Configuration Register (TRAMMCR)
$YFFB02	TPURAM Test Register (TRAMTST)
$YFFB04	TPURAM Base Address and Status Register (TRAMBAR)
$YFFB06 – $YFFB3F	Not Used

NOTES:

[1] Y = M111, where M is the logic state of the module mapping (MM) bit in the SIMCR.

D.9.1 TPURAM Module Configuration Register

TRAMMCR — TPURAM Module Configuration Register **$YFFB00**

15	14	13	12	11	10	9	8	7 0
STOP	0	0	0	0	0	0	RASP	NOT USED
RESET:								
0	0	0	0	0	0	0	1	

STOP — Low-Power Stop Mode Enable

0 = TPURAM operates normally.

1 = TPURAM enters low-power stop mode.

This bit controls whether TPURAM operates normally or enters low-power stop mode. In low-power stop mode, the array retains its contents, but cannot be read or written.

RASP — TPURAM Array Space

0 = TPURAM is accessible in supervisor or user space.

1 = TPURAM is accessible in supervisor space only.

D.9.2 TPURAM Test Register

TRAMTST — TPURAM Test Register **$YFFB02**

Used for factory test only.

D.9.3 TPURAM Module Configuration Register

TRAMBAR — TPURAM Base Address and Status Register **$YFFB04**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR 23	ADDR 22	ADDR 21	ADDR 20	ADDR 19	ADDR 18	ADDR 17	ADDR 16	ADDR 15	ADDR 14	ADDR 13	ADDR 12	0	0	0	RAMDS
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0