



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Not For New Designs
Core Processor	CPU32
Core Size	32-Bit Single-Core
Speed	25MHz
Connectivity	EBI/EMI, SCI, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	15
Program Memory Size	-
Program Memory Type	ROMIess
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	132-BQFP Bumpered
Supplier Device Package	132-PQFP (24.13x24.13)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68332aveh25

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong



1.1 Features

- Central Processing Unit (CPU32)
 - 32-Bit Architecture
 - Virtual Memory Implementation
 - Table Lookup and Interpolate Instruction
 - Improved Exception Handling for Controller Applications
 - High-Level Language Support
 - Background Debugging Mode
 - Fully Static Operation
- 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
- Time Processor Unit (TPU)
 - Dedicated Microengine Operating Independently of CPU32
 - 16 Independent, Programmable Channels and Pins
 - Any Channel can Perform any Time Function
 - Two Timer Count Registers with Programmable Prescalers
 - Selectable Channel Priority Levels
- Queued Serial Module (QSM)
 - Enhanced Serial Communication Interface
 - Queued Serial Peripheral Interface
 - One 8-Bit Dual Function Port
- Static RAM Module with TPU Emulation Capability (TPURAM)
 - 2-Kbytes of Static RAM
 - May be Used as Normal RAM or TPU Microcode Emulation RAM





332 144-PIN QFP





1.4 Address Map

The following figure is a map of the MCU internal addresses. The RAM array is positioned by the base address registers in the associated RAM control block. Unimplemented blocks are mapped externally.



Figure 4 MCU Address Map

1.5 Intermodule Bus

The intermodule bus (IMB) is a standardized bus developed to facilitate both design and operation of modular microcontrollers. It contains circuitry to support exception processing, address space partitioning, multiple interrupt levels, and vectored interrupts. The standardized modules in the MCU communicate with one another and with external components through the IMB. The IMB in the MCU uses 24 address and 16 data lines.



2 Signal Descriptions

2.1 Pin Characteristics

The following table shows MCU pins and their characteristics. All inputs detect CMOS logic levels. All inputs can be put in a high-impedance state, but the method of doing this differs depending upon pin function. Refer to the table, MCU Driver Types, for a description of output drivers. An entry in the discrete I/O column of the MCU Pin Characteristics table indicates that a pin has an alternate I/O function. The port designation is given when it applies. Refer to the MCU Block Diagram for information about port organization.

Pin Mnemonic	Output Driver	Input Synchronized	Input Hysteresis	Discrete I/O	Port Designation
ADDR23/CS10/ECLK	Α	Y	N	0	—
ADDR[22:19]/CS[9:6]	Α	Y	N	0	PC[6:3]
ADDR[18:0]	A	Y	N	_	—
ĀS	В	Y	N	I/O	PE5
AVEC	В	Y	N	I/O	PE2
BERR	В	Y	N	—	—
BG/CS1	В		_		—
BGACK/CS2	В	Y	Ν	_	_
BKPT/DSCLK	—	Y	Y	—	—
BR/CS0	В	Y	N	_	—
CLKOUT	Α		—		—
CSBOOT	В		—	_	—
DATA[15:0] ¹	Aw	Y	Ν		_
DS	В	Y	N	I/O	PE4
DSACK1	В	Y	N	I/O	PE1
DSACKO	В	Y	Ν	I/O	PE0
DSI/IFETCH	А	Y	Y	—	—
DSO/IPIPE	А	—	—	—	—
EXTAL ²			Special	—	—
FC[2:0]/CS[5:3]	А	Y	Ν	0	PC[2:0]
FREEZE/QUOT	А		—	_	—
HALT	Bo	Y	Ν	—	—
IRQ[7:1]	В	Y	Y	I/O	PF[7:1]
MISO	Во	Y	Y	I/O	PQS0
MODCLK ¹	В	Y	N	I/O	PF0
MOSI	Во	Y	Y	I/O	PQS1
PCS0/SS	Во	Y	Y	I/O	PQS3
PCS[3:1]	Bo	Y	Y	I/O	PQS[6:4]
R/W	Α	Y	N	_	—
RESET	Во	Y	Y	—	—
RMC	В	Y	N	I/O	PE3
RXD		N	N		
SCK	Во	Y	Y	I/O	PQS2
SIZ[1:0]	В	Y	Ν	I/O	PE[7:6]

Table 2 MCU Pin Characteristic



3.2.3 Bus Monitor

The internal bus monitor checks for excessively long DSACK response times during normal bus cycles and for excessively long DSACK or AVEC response times during interrupt acknowledge cycles. The monitor asserts BERR if response time is excessive.

DSACK and AVEC response times are measured in clock cycles. The maximum allowable response time can be selected by setting the BMT field.

The monitor does not check DSACK response on the external bus unless the CPU initiates the bus cycle. The BME bit in the SYPCR enables the internal bus monitor for internal to external bus cycles. If a system contains external bus masters, an external bus monitor must be implemented and the internal to external bus monitor option must be disabled.

3.2.4 Halt Monitor

The halt monitor responds to an assertion of \overline{HALT} on the internal bus. A flag in the reset status register (RSR) indicates that the last reset was caused by the halt monitor. The halt monitor reset can be inhibited by the HME bit in the SYPCR.

3.2.5 Spurious Interrupt Monitor

The spurious interrupt monitor issues **BERR** if no interrupt arbitration occurs during an interrupt-acknowledge cycle.

3.2.6 Software Watchdog

The software watchdog is controlled by SWE in the SYPCR. Once enabled, the watchdog requires that a service sequence be written to SWSR on a periodic basis. If servicing does not take place, the watchdog times out and issues a reset. This register can be written at any time, but returns zeros when read.

SWSR — Software Service Register								\$YF	FFA27
15	8	7	6	5	4	3	2	1	0
NOT USED		0	0	0	0	0	0	0	0
RESET:									
		0	0	0	0	0	0	0	0

Register shown with read value

Perform a software watchdog service sequence as follows:

- a. Write \$55 to SWSR.
- b. Write \$AA to SWSR.

Both writes must occur before time-out in the order listed, but any number of instructions can be executed between the two writes.

The watchdog clock rate is affected by SWP and SWT in SYPCR. When SWT[1:0] are modified, a watchdog service sequence must be performed before the new time-out period takes effect.

The reset value of SWP is affected by the state of the MODCLK pin on the rising edge of reset, as shown in the following table.

MODCLK	SWP
0	1
1	0



3.3.3 Clock Control

The clock control circuits determine system clock frequency and clock operation under special circumstances, such as following loss of synthesizer reference or during low-power operation. Clock source is determined by the logic state of the MODCLK pin during reset.

SYNCR	-Clo	ck Syn	thesize	er Cont	trol Re	gister								\$YF	FA04
15	14	13					8	7	6	5	4	3	2	1	0
W	Х			`	Y			EDIV	0	0	SLIMP	SLOCK	RSTEN	STSIM	STEXT
RESET:															
0	0	1	1	1	1	1	1	0	0	0	U	U	0	0	0

When the on-chip clock synthesizer is used, system clock frequency is controlled by the bits in the upper byte of SYNCR. Bits in the lower byte show status of or control operation of internal and external clocks. The SYNCR can be read or written only when the CPU is operating at the supervisor privilege level.

W — Frequency Control (VCO)

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

X — Frequency Control Bit (Prescale)

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. There is no VCO relock delay.

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. Values range from 0 to 63. 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 pin ADDR23. Refer to **3.5 Chip Selects** for more information.

SLIMP — Limp Mode Flag

0 = External crystal is VCO reference.

1 = Loss of crystal reference.

When the on-chip synthesizer is used, loss of reference frequency causes SLIMP to be set. The VCO continues to run using the base control voltage. Maximum limp frequency is maximum specified system clock frequency. X-bit state affects limp frequency.

SLOCK — Synthesizer Lock Flag

0 = VCO is enabled, but has not locked.

1 = VCO has locked on the desired frequency (or system clock is external).

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

RSTEN — Reset Enable

- 0 = Loss of crystal causes the MCU to operate in limp mode.
- 1 = Loss of crystal causes system reset.

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.

STEXT — Stop Mode External Clock

- 0 = When LPSTOP is executed, the CLKOUT signal is held negated to conserve power.
- 1 = When LPSTOP is executed, the CLKOUT signal is driven from the SIM clock, as determined by the state of the STSIM bit.



3.4 External Bus Interface

The external bus interface (EBI) transfers information between the internal MCU bus and external devices. The external bus has 24 address lines and 16 data lines.

The EBI provides dynamic sizing between 8-bit and 16-bit data accesses. It supports byte, word, and long-word transfers. Ports are accessed through the use of asynchronous cycles controlled by the data transfer (SIZ1 and SIZ0) and data size acknowledge pins (DSACK1 and DSACK0). Multiple bus cycles may be required for a transfer to or from an 8-bit port.

Port width is the maximum number of bits accepted or provided during a bus transfer. External devices must follow the handshake protocol described below. Control signals indicate the beginning of the cycle, the address space, the size of the transfer, and the type of cycle. The selected device controls the length of the cycle. Strobe signals, one for the address bus and another for the data bus, indicate the validity of an address and provide timing information for data. The EBI operates in an asynchronous mode for any port width.

To add flexibility and minimize the necessity for external logic, MCU chip-select logic can be synchronized with EBI transfers. Chip-select logic can also provide internally-generated bus control signals for these accesses. Refer to **3.5 Chip Selects** for more information.

3.4.1 Bus Control Signals

The CPU initiates a bus cycle by driving the address, size, function code, and read/write outputs. At the beginning of the cycle, size signals SIZ0 and SIZ1 are driven along with the function code signals. The size signals indicate the number of bytes remaining to be transferred during an operand cycle. They are valid while the address strobe (\overline{AS}) is asserted. The following table shows SIZ0 and SIZ1 encoding. The read/write (R/W) signal determines the direction of the transfer during a bus cycle. This signal changes state, when required, at the beginning of a bus cycle, and is valid while \overline{AS} is asserted. R/W only changes state when a write cycle is preceded by a read cycle or vice versa. The signal can remain low for two consecutive write cycles.

SIZ1	SIZ0	Transfer Size
0	1	Byte
1	0	Word
1	1	Three Byte
0	0	Long Word

Table 8 Size Signal Encoding

3.4.2 Function Codes

The CPU32 automatically generates function code signals FC[2:0]. The function codes can be considered address extensions that automatically select one of eight address spaces to which an address applies. These spaces are designated as either user or supervisor, and program or data spaces. Address space 7 is designated CPU space. CPU space is used for control information not normally associated with read or write bus cycles. Function codes are valid while \overline{AS} is asserted.



CSORB	T —Cł	nip-Se	lect Op	tion R	egister	Boot F	ROM							\$YF	FA4A
15	14	13	12	11	10	9			6	5	4	3		1	0
MODE	BY	TE	R/	W	STRB		DS/	ACK		SP	ACE		IPL		AVEC
RESET:			-												
0	1	1	1	1	0	1	1	0	1	1	1	0	0	0	0
CSOR[1	0:0] —	-Chip-	Select	Optio	n Regis	ters						\$	YFFA4	E-\$YI	FFA76
15	14	13	12	11	10	9			6	5	4	3		1	0
MODE	BY	TE	R/	W	STRB		DS/	ACK		SP	ACE		IPL		AVEC
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSORBT, the option register for CSBOOT, contains special reset values that support bootstrap operations from peripheral memory devices.

The following bit descriptions apply to both CSORBT and CSOR[10:0] option registers.

MODE — Asynchronous/Synchronous Mode

- 0 = Asynchronous mode selected (chip-select assertion determined by internal or external bus control signals)
- 1 = Synchronous mode selected (chip-select assertion synchronized with ECLK signal)

In asynchronous mode, the chip select is asserted synchronized with $\overline{\text{AS}}$ or $\overline{\text{DS}}$.

The DSACK field is not used in synchronous mode because a bus cycle is only performed as a synchronous operation. When a match condition occurs on a chip select programmed for synchronous operation, the chip select signals the EBI that an ECLK cycle is pending.

BYTE — Upper/Lower Byte Option

This field is used only when the chip-select 16-bit port option is selected in the pin assignment register. The following table lists upper/lower byte options.

Byte	Description
00	Disable
01	Lower Byte
10	Upper Byte
11	Both Bytes

R/\overline{W} — Read/Write

This field causes a chip select to be asserted only for a read, only for a write, or for both read and write. Refer to the following table for options available.

R/W	Description
00	Reserved
01	Read Only
10	Write Only
11	Read/Write

STRB — Address Strobe/Data Strobe

0 = Address strobe

1 = Data strobe

This bit controls the timing for assertion of a chip select in asynchronous mode. Selecting address strobe causes chip select to be asserted synchronized with address strobe. Selecting data strobe causes chip select to be asserted synchronized with data strobe.



PEPAR — Port E Pin Assignment Register								\$YI	FA17
15	8	7	6	5	4	3	2	1	0
NOT USED		PEPA7	PEPA6	PEPA5	PEPA4	PEPA3	PEPA2	PEPA1	PEPA0
RESET:									•

DATA8 DATA8 DATA8 DATA8 DATA8 DATA8 DATA8 DATA8

The bits in this register control the function of each port E pin. Any bit set to one configures the corresponding pin as a bus control signal, with the function shown in the following table. Any bit cleared to zero defines the corresponding pin to be an I/O pin, controlled by PORTE and DDRE.

Data bus bit 8 controls the state of this register following reset. If DATA8 is set to one during reset, the register is set to \$FF, which defines all port E pins as bus control signals. If DATA8 is cleared to zero during reset, this register is set to \$00, configuring all port E pins as I/O pins.

Any bit cleared to zero defines the corresponding pin to be an I/O pin. Any bit set to one defines the corresponding pin to be a bus control signal.

PEPAR Bit	Port E Signal	Bus Control Signal
PEPA7	PE7	SIZ1
PEPA6	PE6	SIZO
PEPA5	PE5	ĀS
PEPA4	PE4	DS
PEPA3	PE3	RMC
PEPA2	PE2	AVEC
PEPA1	PE1	DSACK1
PEPA0	PE0	DSACK0

Table 16 Port E Pin Assignments

PORTF0, PORTF1 — Port F Data Register						\$`	YFFA1	9, \$YF	FA1B
15	8	7	6	5	4	3	2	1	0
NOT USED		PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
RESET:									
		U	U	U	U	U	U	U	U

The write to the port F data register is stored in the internal data latch, and if any port F pin is configured as an output, the value stored for that bit is driven onto the pin. A read of the port F data register returns the value at the pin only if the pin is configured as a discrete input. Otherwise, the value read is the value stored in the register.

The port F data register is a single register that can be accessed in two locations. When accessed at \$YFFA19, the register is referred to as PORTF0; when accessed at \$YFFA1B, the register is referred to as PORTF1. The register can be read or written at any time. It is unaffected by reset.

DDRF — Port F Data Direction Register								\$YF	FA1D
15	8	7	6	5	4	3	2	1	0
NOT USED		DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0
RESET:									
		0	0	0	0	0	0	0	0

The bits in this register control the direction of the pin drivers when the pins are configured for I/O. Any bit in this register set to one configures the corresponding pin as an output. Any bit in this register cleared to zero configures the corresponding pin as an input.



3.7.3 Reset Timing

The RESET input must be asserted for a specified minimum period in order for reset to occur. External RESET assertion can be delayed internally for a period equal to the longest bus cycle time (or the bus monitor time-out period) in order to protect write cycles from being aborted by reset. While RESET is asserted, SIM pins are either in a disabled high-impedance state or are driven to their inactive states.

When an external device asserts **RESET** for the proper period, reset control logic clocks the signal into an internal latch. The control logic drives the **RESET** pin low for an additional 512 CLKOUT cycles after it detects that the **RESET** signal is no longer being externally driven, to guarantee this length of reset to the entire system.

If an internal source asserts a reset signal, the reset control logic asserts **RESET** for a minimum of 512 cycles. If the reset signal is still asserted at the end of 512 cycles, the control logic continues to assert **RESET** until the internal reset signal is negated.

After 512 cycles have elapsed, the reset input pin goes to an inactive, high-impedance state for ten cycles. At the end of this 10-cycle period, the reset input is tested. When the input is at logic level one, reset exception processing begins. If, however, the reset input is at logic level zero, the reset control logic drives the pin low for another 512 cycles. At the end of this period, the pin again goes to high-impedance state for ten cycles, then it is tested again. The process repeats until **RESET** is released.

3.7.4 Power-On Reset

When the SIM clock synthesizer is used to generate the system clock, power-on reset involves special circumstances related to application of system and clock synthesizer power. Regardless of clock source, voltage must be applied to clock synthesizer power input pin V_{DDSYN} in order for the MCU to operate. The following discussion assumes that V_{DDSYN} is applied before and during reset. This minimizes crystal start-up time. When V_{DDSYN} is applied at power-on, start-up time is affected by specific crystal parameters and by oscillator circuit design. V_{DD} ramp-up time also affects pin state during reset.

During power-on reset, an internal circuit in the SIM drives the internal (IMB) and external reset lines. The circuit releases the internal reset line as V_{DD} ramps up to the minimum specified value, and SIM pins are initialized. When V_{DD} reaches the specified minimum value, the clock synthesizer VCO begins operation. Clock frequency ramps up to the specified limp mode frequency. The external RESET line remains asserted until the clock synthesizer PLL locks and 512 CLKOUT cycles elapse.

The SIM clock synthesizer provides clock signals to the other MCU modules. After the clock is running and the internal reset signal is asserted for four clock cycles, these modules reset. V_{DD} ramp time and VCO frequency ramp time determine how long these four cycles take. Worst case is approximately 15 milliseconds. During this period, module port pins may be in an indeterminate state. While input-only pins can be put in a known state by means of external pull-up resistors, external logic on input/output or output-only pins must condition the lines during this time. Active drivers require high-impedance buffers or isolation resistors to prevent conflict.

3.7.5 Use of Three State Control Pin

Asserting the three-state control (TSC) input causes the MCU to put all output drivers in an inactive, high-impedance state. The signal must remain asserted for ten clock cycles in order for drivers to change state. There are certain constraints on use of TSC during power-on reset:

When the internal clock synthesizer is used (MODCLK held high during reset), synthesizer rampup time affects how long the ten cycles take. Worst case is approximately 20 milliseconds from TSC assertion.

When an external clock signal is applied (MODCLK held low during reset), pins go to high-impedance state as soon after TSC assertion as ten clock pulses have been applied to the EXTAL pin.



When TSC assertion takes effect, internal signals are forced to values that can cause inadvertent mode selection. Once the output drivers change state, the MCU must be powered down and restarted before normal operation can resume.

3.8 Interrupts

Interrupt recognition and servicing involve complex interaction between the central processing unit, the system integration module, and a device or module requesting interrupt service.

The CPU32 provides for eight levels of interrupt priority (0–7), seven automatic interrupt vectors, and 200 assignable interrupt vectors. All interrupts with priorities less than seven can be masked by the interrupt priority (IP) field in the status register. The CPU32 handles interrupts as a type of asynchronous exception.

Interrupt recognition is based on the states of interrupt request signals <u>iIRQ[7:1]</u> and the IP mask value. Each of the signals corresponds to an interrupt priority. <u>IRQ1</u> has the lowest priority, and <u>IRQ7</u> has the highest priority.

The IP field consists of three bits. Binary values %000 to %111 provide eight priority masks. Masks prevent an interrupt request of a priority less than or equal to the mask value (except for IRQ7) from being recognized and processed. When IP contains %000, no interrupt is masked. During exception processing, the IP field is set to the priority of the interrupt being serviced.

Interrupt request signals can be asserted by external devices or by microcontroller modules. Request lines are connected internally by means of a wired NOR — simultaneous requests of differing priority can be made. Internal assertion of an interrupt request signal does not affect the logic state of the corresponding MCU pin.

External interrupt requests are routed to the CPU via the external bus interface and SIM interrupt control logic. The CPU treats external interrupt requests as though they come from the SIM.

External IRQ[6:1] are active-low level-sensitive inputs. External IRQ7 is an active-low transition-sensitive input. IRQ7 requires both an edge and a voltage level for validity.

IRQ[6:1] are maskable. IRQ7 is nonmaskable. The IRQ7 input is transition-sensitive in order to prevent redundant servicing and stack overflow. A nonmaskable interrupt is generated each time IRQ7 is asserted, and each time the priority mask changes from %111 to a lower number while IRQ7 is asserted.

Interrupt requests are sampled on consecutive falling edges of the system clock. Interrupt request input circuitry has hysteresis. To be valid, a request signal must be asserted for at least two consecutive clock periods. Valid requests do not cause immediate exception processing, but are left pending. Pending requests are processed at instruction boundaries or when exception processing of higher-priority exceptions is complete.

The CPU32 does not latch the priority of a pending interrupt request. If an interrupt source of higher priority makes a service request while a lower priority request is pending, the higher priority request is serviced. If an interrupt request of equal or lower priority than the current IP mask value is made, the CPU does not recognize the occurrence of the request in any way.

3.8.1 Interrupt Acknowledge and Arbitration

Interrupt acknowledge bus cycles are generated during exception processing. When the CPU detects one or more interrupt requests of a priority higher than the interrupt priority mask value, it performs a CPU space read from address \$FFFFF : [IP] : 1.

The CPU space read cycle performs two functions: it places a mask value corresponding to the highest priority interrupt request on the address bus, and it acquires an exception vector number from the interrupt source. The mask value also serves two purposes: it is latched into the CCR IP field in order to



Instruction	Syntax	Operand Size	Operation
DBcc	Dn, label	16	If condition false, then $Dn - 1 \Rightarrow PC$; if $Dn \neq (-1)$, then $PC + d \Rightarrow PC$
DIVS/DIVU	<ea>, Dn</ea>	32/16 ⇒ 16 : 16	Destination / Source \Rightarrow Destination (signed or unsigned)
DIVSL/DIVUL	<ea>, Dr : Dq <ea>, Dq <ea>, Dr : Dq</ea></ea></ea>	$\begin{array}{c} 64/32 \Rightarrow 32:32\\ 32/32 \Rightarrow 32\\ 32/32 \Rightarrow 32:32 \end{array}$	Destination / Source \Rightarrow Destination (signed or unsigned)
EOR	Dn, <ea></ea>	8, 16, 32	Source \oplus Destination \Rightarrow Destination
EORI	# <data>, <ea></ea></data>	8, 16, 32	Data \oplus Destination \Rightarrow Destination
EORI to CCR	# <data>, CCR</data>	8	Source \oplus CCR \Rightarrow CCR
EORI to SR ¹	# <data>, SR</data>	16	Source \oplus SR \Rightarrow SR
EXG	Rn, Rn	32	$Rn \Rightarrow Rn$
EXT	Dn Dn	$ \begin{array}{c} 8 \Rightarrow 16 \\ 16 \Rightarrow 32 \end{array} $	Sign extended Destination \Rightarrow Destination
EXTB	Dn	8 ⇒ 32	Sign extended Destination \Rightarrow Destination
ILLEGAL	none	none	$\begin{array}{l} \text{SSP}-2\Rightarrow \text{SSP}; \text{ vector offset} \Rightarrow (\text{SSP});\\ \text{SSP}-4\Rightarrow \text{SSP}; \text{PC} \Rightarrow (\text{SSP});\\ \text{SSP}-2\Rightarrow \text{SSP}; \text{SR} \Rightarrow (\text{SSP});\\ \text{Illegal instruction vector address} \Rightarrow \text{PC} \end{array}$
JMP	Í	none	$Destination \Rightarrow PC$
JSR	Í	none	SP – 4 \Rightarrow SP; PC \Rightarrow (SP); destination \Rightarrow PC
LEA	<ea>, An</ea>	32	$\langle ea \rangle \Rightarrow An$
LINK	An, # d	16, 32	$SP-4\RightarrowSP,An\Rightarrow(SP);SP\RightarrowAn,SP+d\RightarrowSP$
LPSTOP ¹	# <data></data>	16	Data \Rightarrow SR; interrupt mask \Rightarrow EBI; STOP
LSL	Dn, Dn # <data>, Dn Í</data>	8, 16, 32 8, 16, 32 16	X/C - 0
LSR	Dn, Dn # <data>, Dn Í</data>	8, 16, 32 8, 16, 32 16	0 → X/C
MOVE	<ea>, <ea></ea></ea>	8, 16, 32	Source \Rightarrow Destination
MOVEA	<ea>, An</ea>	16, 32 ⇒ 32	Source \Rightarrow Destination
MOVEA ¹	USP, An An, USP	32 32	$\begin{array}{l} \text{USP} \Rightarrow \text{An} \\ \text{An} \Rightarrow \text{USP} \end{array}$
MOVE from CCR	CCR, <ea></ea>	16	$CCR \Rightarrow Destination$
MOVE to CCR	<ea>, CCR</ea>	16	Source \Rightarrow CCR
MOVE from SR ¹	SR, <ea></ea>	16	$SR \Rightarrow Destination$
MOVE to SR ¹	<ea>, SR</ea>	16	Source \Rightarrow SR
MOVE USP ¹	USP, An An, USP	32 32	$\begin{array}{l} USP \Rightarrow An \\ An \Rightarrow USP \end{array}$
MOVEC ¹	Rc, Rn Rn, Rc	32 32	$ \begin{array}{l} Rc \Rightarrow Rn \\ Rn \Rightarrow Rc \end{array} $
MOVEM	list, <ea> <ea>, list</ea></ea>	16, 32 16, 32 ⇒ 32	Listed registers \Rightarrow Destination Source \Rightarrow Listed registers
MOVEP	Dn, (d16, An)	16, 32	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$
	(d16, An), Dn		$\begin{array}{l} (An+d) \Rightarrow Dn \ [31:24]; \ (An+d+2) \Rightarrow Dn \ [23:16]; \\ (An+d+4) \Rightarrow Dn \ [15:8]; \ (An+d+6) \Rightarrow Dn \ [7:0] \end{array}$
MOVEQ	# <data>, Dn</data>	$8 \Rightarrow 32$	Immediate data \Rightarrow Destination

Table 20 Instruction Set Summary(Continued)



		1	
Instruction	Syntax	Operand Size	Operation
MOVES ¹	Rn, <ea> <ea>, Rn</ea></ea>	8, 16, 32	$Rn \Rightarrow$ Destination using DFC Source using SFC \Rightarrow Rn
MULS/MULU	<ea>, Dn <ea>, Dl <ea>, Dh : Dl</ea></ea></ea>	$16 * 16 \Rightarrow 32$ $32 * 32 \Rightarrow 32$ $32 * 32 \Rightarrow 64$	Source $*$ Destination \Rightarrow Destination (signed or unsigned)
NBCD	Í	8 8	$0 - \text{Destination}_{10} - X \Rightarrow \text{Destination}$
NEG	Í	8, 16, 32	$0 - Destination \Rightarrow Destination$
NEGX	Í	8, 16, 32	$0 - Destination - X \Rightarrow Destination$
NOP	none	none	$PC + 2 \Rightarrow PC$
NOT	Í	8, 16, 32	$\overline{\text{Destination}} \Rightarrow \text{Destination}$
OR	<ea>, Dn Dn, <ea></ea></ea>	8, 16, 32 8, 16, 32	Source + Destination \Rightarrow Destination
ORI	# <data>, <ea></ea></data>	8, 16, 32	Data + Destination \Rightarrow Destination
ORI to CCR	# <data>, CCR</data>	16	Source + CCR \Rightarrow SR
ORI to SR ¹	# <data>, SR</data>	16	Source ; SR \Rightarrow SR
PEA	Í	32	$SP - 4 \Rightarrow SP; \langle ea \rangle \Rightarrow SP$
RESET ¹	none	none	Assert RESET line
ROL	Dn, Dn # <data>, Dn Í</data>	8, 16, 32 8, 16, 32 16	
ROR	Dn, Dn # <data>, Dn Í</data>	8, 16, 32 8, 16, 32 16	
ROXL	Dn, Dn # <data>, Dn Í</data>	8, 16, 32 8, 16, 32 16	
ROXR	Dn, Dn # <data>, Dn Í</data>	8, 16, 32 8, 16, 32 16	
RTD	#d	16	$(SP) \Rightarrow PC; SP + 4 + d \Rightarrow SP$
RTE ¹	none	none	$(SP) \Rightarrow SR; SP + 2 \Rightarrow SP; (SP) \Rightarrow PC;$ SP + 4 \Rightarrow SP; Restore stack according to format
RTR	none	none	$(SP) \Rightarrow CCR; SP + 2 \Rightarrow SP; (SP) \Rightarrow PC; SP + 4 \Rightarrow SP$
RTS	none	none	$(SP) \Rightarrow PC; SP + 4 \Rightarrow SP$
SBCD	Dn, Dn – (An), – (An)	8 8	Destination10 – Source10 – $X \Rightarrow$ Destination
Scc	Í	8	If condition true, then destination bits are set to 1; else, destination bits are cleared to 0
STOP ¹	# <data></data>	16	Data \Rightarrow SR; STOP
SUB	<ea>, Dn Dn, <ea></ea></ea>	8, 16, 32	Destination – Source \Rightarrow Destination
SUBA	<ea>, An</ea>	16, 32	Destination – Source \Rightarrow Destination
SUBI	# <data>, <ea></ea></data>	8, 16, 32	Destination – Data \Rightarrow Destination
SUBQ	# <data>, <ea></ea></data>	8, 16, 32	Destination – Data \Rightarrow Destination
SUBX	Dn, Dn – (An), – (An)	8, 16, 32 8, 16, 32	Destination – Source – $X \Rightarrow$ Destination

Table 20 Instruction Set Summary(Continued)



5.2.3 Queued Output Match (QOM)

QOM can generate single or multiple output match events from a table of offsets in parameter RAM. Loop modes allow complex pulse trains to be generated once, a specified number of times, or continuously. The function can be triggered by a link from another TPU channel. In addition, the reference time for the sequence of matches can be obtained from another channel. QOM can generate pulse-width modulated waveforms, including waveforms with high times of 0% or 100%. QOM also allows a TPU channel to be used as a discrete output pin.

5.2.4 Programmable Time Accumulator (PTA)

PTA accumulates a 32-bit sum of the total high time, low time, or period of an input signal over a programmable number of periods or pulses. The accumulation can start on a rising or falling edge. After the specified number of periods or pulses, the PTA generates an interrupt request and optionally generates links to other channels.

From 1 to 255 period measurements can be made and summed with the previous measurement(s) before the TPU interrupts the CPU, providing instantaneous or average frequency measurement capability, and the latest complete accumulation (over the programmed number of periods).

5.2.5 Multichannel Pulse Width Modulation (MCPWM)

MCPWM generates pulse-width modulated outputs with full 0% to 100% duty cycle range independent of other TPU activity. This capability requires two TPU channels plus an external gate for one PWM channel. (A simple one-channel PWM capability is supported by the QOM function.)

Multiple PWMs generated by MCPWM have two types of high time alignment: edge aligned and center aligned. Edge aligned mode uses n + 1 TPU channels for n PWMs; center aligned mode uses 2n + 1 channels. Center aligned mode allows a user defined 'dead time' to be specified so that two PWMs can be used to drive an H-bridge without destructive current spikes. This feature is important for motor control applications.

5.2.6 Fast Quadrature Decode (FQD)

FQD is a position feedback function for motor control. It decodes the two signals from a slotted encoder to provide the CPU with a 16-bit free running position counter. FQD incorporates a "speed switch" which disables one of the channels at high speed, allowing faster signals to be decoded. A time stamp is provided on every counter update to allow position interpolation and better velocity determination at low speed or when low resolution encoders are used. The third index channel provided by some encoders is handled by the ICTC function.

5.2.7 Universal Asynchronous Receiver/Transmitter (UART)

The UART function uses one or two TPU channels to provide asynchronous communications. Data word length is programmable from 1 to 14 bits. The function supports detection or generation of even, odd, and no parity. Baud rate is freely programmable and can be higher than 100 Kbaud. Eight bidirectional UART channels running in excess of 9600 baud could be implemented on the TPU.

5.2.8 Brushless Motor Commutation (COMM)

This function generates the phase commutation signals for a variety of brushless motors, including three-phase brushless direct current. It derives the commutation state directly from the position decoded in FQD, thus eliminating the need for hall effect sensors.

The state sequence is implemented as a user-configurable state machine, thus providing a flexible approach with other general applications. A CPU offset parameter is provided to allow all the switching angles to be advanced or retarded on the fly by the CPU. This feature is useful for torque maintenance at high speeds.



5.2.9 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.

5.2.10 Hall Effect Decode (HALLD)

This function decodes the sensor signals from a brushless motor, along with a direction input from the CPU, 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.

5.3 Programmer's Model

The TPU control register address map occupies 512 bytes. The "Access" column in the TPU address map below indicates which registers are accessible only at the supervisor privilege level and which can be assigned to either the supervisor or user privilege level, according to the value of the SUPV bit in the TPUMCR.

Access	Address	15 8 7	0
S	\$YFFE00	TPU 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)	

Table 22 TPU Address Map

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



5.4 Parameter RAM

Parameter RAM occupies 256 bytes at the top of the TPU module address map. Channel parameters are organized as 128 16-bit words. However, only 100 words are actually implemented. The parameter RAM address map shows how parameter words are organized in memory.

Channel	Base	Parameter Address								
Number	Address	0	1	2	3	4	5	6	7	
0	\$YFFFF##	00	02	04	06	08	0A	—	—	
1	\$YFFFF##	10	12	14	16	18	1A	_	_	
2	\$YFFFF##	20	22	24	26	28	2A	_	_	
3	\$YFFFF##	30	32	34	36	38	ЗA	_	_	
4	\$YFFFF##	40	42	44	46	48	4A	_	_	
5	\$YFFFF##	50	52	54	56	58	5A	_	_	
6	\$YFFFF##	60	62	64	66	68	6A	_	_	
7	\$YFFFF##	70	72	74	76	78	7A	_	_	
8	\$YFFFF##	80	82	84	86	88	8A	_	_	
9	\$YFFFF##	90	92	94	96	98	9A	_	_	
10	\$YFFFF##	A0	A2	A4	A6	A8	AA	_	_	
11	\$YFFFF##	B0	B2	B4	B6	B8	BA	_	_	
12	\$YFFFF##	C0	C2	C4	C6	C8	CA	_	_	
13	\$YFFFF##	D0	D2	D4	D6	D8	DA	_	_	
14	\$YFFFF##	E0	E2	E4	E6	E8	EA	EC	EE	
15	\$YFFFF##	F0	F2	F4	F6	F8	FA	FC	FE	

Table 23 TPU Parameter RAM Address Map

--= Not Implemented

Y = M111, where M represents the logic state of the MM bit in the SIMCR.

5.5 TPU Registers

The TPU memory map contains three groups of registers:

System Configuration Registers Channel Control and Status Registers Development Support and Test Verification Registers

5.5.1 System Configuration Registers

	TPUMCR —	- TPU Module	Configuration	Register
--	----------	--------------	---------------	----------

15	14	13	12	11	10	9	8	7	6	5	4	3	0		
STOP	TCF	R1P	TCF	R2P	EMU	T2CG	STF	SUPV	PSCK	0	0	IARB			
RESET:															
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

STOP — Stop Bit

0 = TPU operating normally

1 = Internal clocks shut down

\$YFFE00



QSM Pin	Mode	DDRQS	Bit	Pin Function
		Bit	State	
MISO	Master	DDQ0	0	Serial Data Input to QSPI
			1	Disables Data Input
	Slave		0	Disables Data Output
			1	Serial Data Output from QSPI
MOSI	Master	DDQ1	0	Disables Data Output
			1	Serial Data Output from QSPI
	Slave		0	Serial Data Input to QSPI
			1	Disables Data Input
SCK ¹	Master	DDQ2	0	Disables Clock Output
			1	Clock Output from QSPI
	Slave		0	Clock Input to QSPI
			1	Disables Clock Input
PCS0/SS	Master	DDQ3	0	Assertion Causes Mode Fault
			1	Chip-Select Output
	Slave		0	QSPI Slave Select Input
			1	Disables Select Input
PCS[3:1]	Master	DDQ[4:6]	0	Disables Chip-Select Output
			1	Chip-Select Output
	Slave		0	Inactive
			1	Inactive
TXD ²	Transmit	DDQ7	Х	Serial Data Output from SCI
RXD	Receive	None	NA	Serial Data Input to SCI

Table 26 Effect of DDRQS on QSM Pin Function

NOTES:

- 1. PQS2 is a digital I/O pin unless the SPI is enabled (SPE in SPCR1 set), in which case it becomes SPI serial clock SCK.
- 2. PQS7 is a digital I/O pin unless the SCI transmitter is enabled (TE in SCCR1 = 1), in which case it becomes SCI serial output TXD.

DDRQS determines the direction of the TXD pin only when the SCI transmitter is disabled. When the SCI transmitter is enabled, the TXD pin is an output.



Pin Names	Mnemonics	Mode	Function
Master In Slave Out	MISO	Master Slave	Serial Data Input to QSPI Serial Data Output from QSPI
Master Out Slave In	MOSI	Master Slave	Serial Data Output from QSPI Serial Data Input to QSPI
Serial Clock	SCK	Master Slave	Clock Output from QSPI Clock Input to QSPI
Peripheral Chip Selects	PCS[3:1]	Master	Select Peripherals
Peripheral Chip Select Slave Select	PCS0 SS	Master Master Slave	Selects Peripheral Causes Mode Fault Initiates Serial Transfer

6.5.2 QSPI Registers

The programmer's model for the QSPI submodule consists of the QSM global and pin control registers, four QSPI control registers, one status register, and the 80-byte QSPI RAM.

The CPU can read and write to registers and RAM. The four control registers must be initialized before the QSPI is enabled to ensure defined operation. SPCR1 should be written last because it contains QSPI enable bit SPE. Asserting this bit starts the QSPI. The QSPI control registers are reset to a defined state and can then be changed by the CPU. Reset values are shown below each register.

Refer to the following memory map of the QSPI.

Address	Name	Usage
\$YFFC18	SPCR0	QSPI Control Register 0
\$YFFC1A	SPCR1	QSPI Control Register 1
\$YFFC1C	SPCR2	QSPI Control Register 2
\$YFFC1E	SPCR3	QSPI Control Register 3
\$YFFC1F	SPSR	QSPI Status Register
\$YFFD00	RAM	QSPI Receive Data (16 Words)
\$YFFD20	RAM	QSPI Transmit Data (16 Words)
\$YFFD40	RAM	QSPI Command Control (8 Words)

Writing a different value into any control register except SPCR2 while the QSPI is enabled disrupts operation. SPCR2 is buffered to prevent disruption of the current serial transfer. After completion of the current serial transfer, the new SPCR2 values become effective.

Writing the same value into any control register except SPCR2 while the QSPI is enabled has no effect on QSPI operation. Rewriting NEWQP in SPCR2 causes execution to restart at the designated location.

SPCR0	— QSI	SPI Control Register 0											\$YFFC18		
15	14	13			10	9	8	7							0
MSTR	WOMQ		Bl	TS		CPOL	CPHA				SP	BR			
RESET:															
0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0

SPCR0 contains parameters for configuring the QSPI before it is enabled. The CPU can read and write this register. The QSM has read-only access.



IDLE — Idle-Line Detected Flag

0 = SCI receiver did not detect an idle-line condition.

1 = SCI receiver detected an idle-line condition.

IDLE is disabled when RWU in SCCR1 is set. IDLE is set when the SCI receiver detects the idle-line condition specified by ILT in SCCR1. If cleared, IDLE will not set again until after RDRF is set. RDRF is set when a break is received, so that a subsequent idle line can be detected.

OR — Overrun Error Flag

0 = RDRF is cleared before new data arrives.

1 = RDRF is not cleared before new data arrives.

OR is set when a new byte is ready to be transferred from the receive serial shifter to the RDR, and RDRF is still set. Data transfer is inhibited until OR is cleared. Previous data in RDR remains valid, but data received during overrun condition (including the byte that set OR) is lost.

NF — Noise Error Flag

- 0 = No noise detected on the received data
- 1 = Noise occurred on the received data

NF is set when the SCI receiver detects noise on a valid start bit, on any data bit, or on a stop bit. It is not set by noise on the idle line or on invalid start bits. Each bit is sampled three times. If none of the three samples are the same logic level, the majority value is used for the received data value, and NF is set. NF is not set until an entire frame is received and RDRF is set.

FE — Framing Error Flag

0 = No framing error on the received data.

1 = Framing error or break occurred on the received data.

FE is set when the SCI receiver detects a zero where a stop bit was to have occurred. FE is not set until the entire frame is received and RDRF is set. A break can also cause FE to be set. It is possible to miss a framing error if RXD happens to be at logic level one at the time the stop bit is expected.

PF — Parity Error Flag

0 = No parity error on the received data

1 = Parity error occurred on the received data

PF is set when the SCI receiver detects a parity error. PF is not set until the entire frame is received and RDRF is set.

SCDR -	– SCI I	Data R	Registe	r										\$YF	FC0E
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	R8/T8	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0
RESET:															
0	0	0	0	0	0	0	U	U	U	U	U	U	U	U	U

SCDR contains two data registers at the same address. Receive data register (RDR) is a read-only register that contains data received by the SCI. The data comes into the receive serial shifter and is transferred to RDR. Transmit data register (TDR) is a write-only register that contains data to be transmitted. The data is first written to TDR, then transferred to the transmit serial shifter, where additional format bits are added before transmission. R[7:0]/T[7:0] contain either the first eight data bits received when SCDR is read, or the first eight data bits to be transmitted when SCDR is written. R8/T8 are used when the SCI is configured for 9-bit operation. When it is configured for 8-bit operation, they have no meaning or effect.



8 Summary of Changes

This is a partial revision. Most of the publication remains the same, but the following changes were made to improve it. Typographical errors that do not affect content are not annotated. This document has also been reformatted for use on the web.

Pages 2-3	New Ordering Information included.
Page 6	New block diagram drawn.
Page 7	New 132-pin assignment diagram drawn.
Page 8	New 144-pin assignment diagram drawn.
Page 9	New address map drawn.
Pages 10-14	Added Signal Description section.
Pages 15-47	Expanded and revised SIM section. Made all register diagrams and bit mnemonics consistent. Incorporated new information concerning the system clock, resets, interrupts, and chip-selects circuits.
Page 48-56	Expanded and revised CPU section. Made all register diagrams and bit mnemon- ics consistent. Revised instruction set summary information.
Page 57-70	Expanded and revised TPU section. Made all register diagrams and bit mnemonics consistent. Revised time functions information to include both MC68332A and MC68332G microcode ROM applications.
Page 71-92	Expanded and revised QSM section. Made all register diagrams and bit mnemon- ics consistent. Added information concerning SPI and SCI operation.
Page 93-95	Revised Standby RAM with TPU Emulation RAM section. Made all register dia- grams and bit mnemonics consistent.