Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Obsolete
Core Processor	CPU32+
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	25MHz
Co-Processors/DSP	Communications; CPM
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	10Mbps (1)
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	0°C ~ 70°C (TA)
Security Features	-
Package / Case	357-BGA
Supplier Device Package	357-PBGA (25x25)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68360zp25lr2

Home Page:
www.freescale.com
email:
support@freescale.com
USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130
support@freescale.com
Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com
Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 2666 8080
support.japan@freescale.com
Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com
For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

field allows the return-from-exception (RTE) instruction to identify what information is on the stack so that it may be properly restored.

5.1.5 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; this flexibility eliminates the need for extra instructions to store register contents in memory.

The seven basic addressing modes are as follows:

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

Included in the register indirect addressing modes are the capabilities to postincrement, predecrement, and offset. The PC relative mode also has index and offset capabilities. In addition to these addressing modes, many instructions implicitly specify the use of the SR, SP and/or PC. Addressing is explained fully in the M68000PM/AD, *M68000 Family Programmer's Reference Manual*.

5.2 ARCHITECTURE SUMMARY

The CPU32+ is upward source- and object-code compatible with the MC68000 and MC68010. It is downward source- and object-code compatible with the MC68020. Within the M68000 family, architectural differences are limited to the supervisory operating state. User programs can be executed unchanged on upward-compatible devices.

The major CPU32+ features are as follows:

- 32-Bit Internal Data Path and Arithmetic Hardware
- 32-Bit Address Bus Supported by 32-Bit Calculations
- Rich Instruction Set
- Eight 32-Bit General-Purpose Data Registers
- Seven 32-Bit General-Purpose Address Registers
- Separate User and Supervisor Stack Pointers (USP and SSP)
- Separate User and Supervisor Address Spaces
- Separate Program and Data Address Spaces
- Many Data Types
- Flexible Addressing Modes
- Full Interrupt Processing
- Expansion Capability

5.3.4.1 TABLE EXAMPLE 1: STANDARD USAGE. The table consists of 257 word entries. As shown in Figure 5-7, the function is linear within the range 32768 ≤ X ≤ 49152. Table entries within this range are as given in Table 5-13 .

Table 5-13. Standard Usage Entries

Entry Number	X-Value	Y-Value
128*	32768	1311
162	41472	1659
163	41728	1669
164	41984	1679
165	42240	1690
192*	49152	1966

*These values are the end points of the range.
All entries between these points fall on the line.

Figure 5-7. Table Example 1

The table instruction is executed with the following bit pattern in Dx:

31 … 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED	1	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0

Table Entry Offset ⇒ Dx [8:15] = $A3 = 163

Interpolation Fraction ⇒ Dx [0:7] = $80 = 128

Using this information, the table instruction calculates dependent variable Y:

Y = 1669 + (128 (1679 – 1669)) / 256 = 1674

In the following equations, negative tail values are used to negate the effects of a slower bus. The equations are generalized, however, so that they may be used on any speed bus with any tail value.

```
NEW_TAIL = OLD_TAIL + (NEW_CLOCK – 2)
IF ((NEW_CLOCK – 4) > 0) THEN
    NEW_CYCLE = OLD_CYCLE + (NEW_CLOCK – 2) + (NEW_CLOCK – 4)
ELSE
    NEW_CYCLE = OLD_CYCLE + (NEW _CLOCK – 2)
```

where:

NEW_TAIL/NEW_CYCLE is the adjusted tail/cycle at the slower speed

OLD_TAIL/OLD_CYCLE is the value listed in the instruction timing tables

NEW_CLOCK is the number of clocks per cycle at the slower speed

Note that many instructions listed as having negative tails are change-of-flow instructions and that the bus speed used in the calculation is that of the new instruction stream.

5.7.2 Instruction Timing Tables

The following assumptions apply to the times shown in the subsequent tables:

1. A 16-bit data bus is used for all memory accesses (CPU32+ in 16-bit mode).
2. Memory access times are based on two-clock bus cycles with no wait states.
3. The instruction pipeline is full at the beginning of the instruction and is refilled by the end of the instruction.

Three values are listed for each instruction and addressing mode:

Head: The number of cycles available at the beginning of an instruction to complete a previous instruction write or to perform a prefetch.

Tail: The number of cycles an instruction uses to complete a write.

Cycles: Four numbers per entry, three contained in parentheses. The outer number is the minimum number of cycles required for the instruction to complete. Numbers within the parentheses represent the number of bus accesses performed by the instruction. The first number is the number of operand read accesses performed by the instruction. The second number is the number of instruction fetches performed by the instruction, including all prefetches that keep the instruction and the instruction pipeline filled. The third number is the number of write accesses performed by the instruction.

As an example, consider an ADD.L (12, A3, D7.W ∗ 4), D2 instruction.

Paragraph 5.7.2.5 Arithmetic/Logic Instructions shows that the instruction has a head = 0, a tail = 0, and cycles = 2 (0/1/0). However, in indexed address register indirect addressing mode, additional time is required to fetch the EA. Paragraph 5.7.2.1 Fetch Effective Address gives addressing mode data. For (d_8, An, Xn.Sz ∗ Scale), head = 4, tail = 2, cycles = 8 (2/1/0). Because this example is for a long access and the fetch EA table lists data for word

NOTE

All accesses to the QUICC internal RAM and registers (including MBAR) by an external master are asynchronous to the QUICC clock. Read and write accesses are with three wait states, and $\overline{\text{DSACK}}$ is asserted by the QUICC assuming three-wait-state accesses.

6.7.1 Initial Configuration

The QUICC has three configuration (CONFIG) pins that are sampled during system (or power-up) reset to select the initial size of the global chip select and whether the QUICC is in the normal (CPU32+ enabled) mode or the slave (CPU32+ disabled) mode (see Table 6-2).

See 6.10 Memory Controller for a description of the global chip select and 6.8 Slave (Disable CPU32+) Mode for a description of slave mode. In normal mode, the global chip select can initially assume the boot ROM port size to be either 8, 16, or 32 bits. In the slave mode, the global chip select can be enabled with 8, 16, or 32 bits, or the global chip select can be disabled. The global chip select would normally be disabled if another QUICC or processor was providing the boot ROM chip select function.

Table 6-2. QUICC Initial Configuration

Configuration Pins			Result
CONFIG2 /FREEZE	CONFIG1 /$\overline{\text{BCLRO}}$	CONFIG0 /RMC	
0	0	0	Slave mode; global chip select 8-bit size; MBAR at $003FF00.
0	0	1	Slave mode; global chip select 32-bit size; MBAR at $003FF00; not MC68040 companion mode; $\overline{\text{BR}}$ output, $\overline{\text{BG}}$ input.
0	1	0	Slave mode; global chip select 16-bit size; MBAR at $003FF00.
0	1	1	MC68040 companion mode; global chip select 32-bit size; MBAR at $003FF00; $\overline{\text{BR}}$ input, $\overline{\text{BG}}$ output.
1	0	0	CPU32+ enabled; global chip select 32-bit size; MBAR at $003FF00.
1	0	1	CPU32+ enabled; global chip select 16-bit size; MBAR at $003FF00.
1	1	0	Slave mode; global chip select disabled; MBAR at $003FF04.
1	1	1	CPU32+ enabled; global chip select 8-bit size; MBAR at $003FF00.

6.7.2 Port D

If the user configures a 16-bit data bus by driving a zero voltage on the PRTY3 pin during system reset, then the D0–D15 pins are not used as a data bus, but are referred to as port D. At this time, port D is not available for general-purpose I/O or any other alternate function on the QUICC. In the future, these pins may be defined to have an additional function in 16-bit data bus mode.

for entering into low-power mode). Clearing this bit switches the general system clock to the DFNH value. CSRC is cleared at hardware reset.

0 = General system clock is determined by the DFNH value.
1 = General system clock is determined by the DFNL value.

6.9.3.12 BREAKPOINT ADDRESS REGISTER (BKAR). This register contains the 32-bit breakpoint address used in the breakpoint address match function. Its contents are only valid if the valid bit is set in the BKCR. BKAR is undefined at reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8	BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

6.9.3.13 BREAKPOINT CONTROL REGISTER (BKCR). This register contains miscellaneous bits required for the breakpoint address match function. BKCR is cleared at reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	BAS	BUSS	RW1	RW0
												0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIZM	SIZ1	SIZ0	NEG	MA1	MA0	AS8	AS7	AS6	AS5	AS4	AS3	AS2	AS1	AS0	V
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 31–20—Reserved

BAS—Breakpoint Acknowledge Support

This bit determines whether to support the CPU32+ breakpoint acknowledge cycle by asserting $\overline{\text{BERR}}$ or ignore the breakpoint acknowledge cycle by allowing it to be handled by the external bus.

0 = No action taken during CPU32+ breakpoint acknowledge cycles.
1 = Assert $\overline{\text{BERR}}$ during CPU32+ breakpoint acknowledge cycles.

NOTE

Do not assert this bit if the QUICC is in slave mode.

BUSS—Bus Select

This bit determines whether the breakpoint logic will use the IMB value or the external bus value to detect breakpoint match.

0 = Use the IMB.
1 = Use the external bus. A0 and A1 are masked from the comparison.

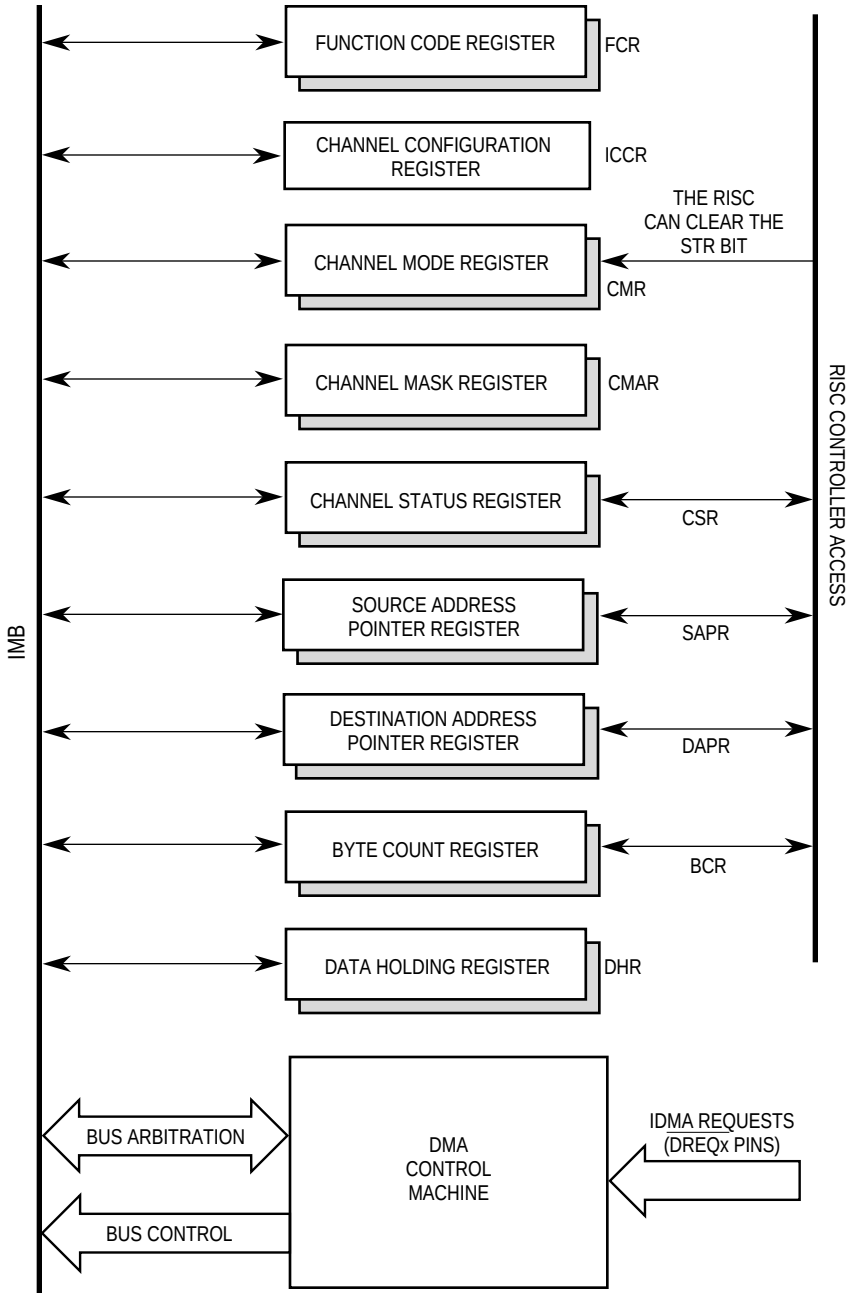

Figure 7-8. IDMA Controller Block Diagram

7.6.1 IDMA Key Features;

The IDMA contains the following features:

- Two Independent, Fully Programmable DMA Channels
- Dual Address or Single Address Transfers with 32-Bit Address and 32-Bit Data Capability

If its V-bit is set, TbPTR contains the address of the Tx entry currently active. The SI RAM transmit address block in use is 192–255, and CROTa = 1 in SISTR.

7.8.5.6.3 SIRP When RDM = 10 (Two Static TDMs). •This is the simplest case, since each pointer is continuously used and has only one function.

RaPTR contains the address of the RXa entry currently active.

RbPTR contains the address of the RXb entry currently active.

TaPTR contains the address of the TXa entry currently active.

TbPTR contains the address of the TXb entry currently active.

7.8.5.6.4 SIRP When RDM = 11 (Two Dynamic TDMs). •In this case, each pointer is continuously used, but points to different sections of the SI RAM, depending on whether the pointer's value is in the first half (0–15) or the second half (16–31).

RaPTR contains the address of the RXa entry currently active. If the pointer has a value from 0–15, the current-route RAM is SI RAM address block 0–31, and CRORa = 0 in SISTR. If the pointer has a value from 16–31, the current-route RAM is SI RAM address block 32–63, and CRORa = 1 in SISTR.

RbPTR contains the address of the RXb entry currently active. If the pointer has a value from 0–15, the current route RAM is SI RAM address block 64–95, and CRORb = 0 in SISTR. If the pointer has a value from 16–31, the current-route RAM is SI RAM address block 96–127, and CRORb = 1 in SISTR.

TaPTR contains the address of the TXa entry currently active. If the pointer has a value from 0–15, the current route RAM is SI RAM address block 128–159, and CROTa = 0 in SISTR. If the pointer has a value from 16–31, the current-route RAM is SI RAM address block 160–191, and CROTa = 1 in SISTR.

TbPTR contains the address of the TXb entry currently active. If the pointer has a value from 0–15, the current-route RAM is SI RAM address block 192–223, and CROTb = 0 in SISTR. If the pointer has a value from 224–255, the current-route RAM is SI RAM address block 160–191, and CROTb = 1 in SISTR.

7.8.6 SI IDL Interface Support

The IDL interface is a full-duplex ISDN interface used to connect a physical layer device to the QUICC. The QUICC supports both the basic rate and the primary rate of the IDL bus. In the basic rate of IDL, data on three channels, B1, B2, and D, is transferred in a 20-bit frame, providing 160-kbps full-duplex bandwidth. The QUICC is an IDL slave device that is clocked by the IDL bus master (physical layer device) and has separate receive and transmit sections. Because the QUICC can support two TDMs, it can actually support two independent IDL buses using separate clocks and sync pulses as shown in Figure 7-31.

receiver is not in hunt mode and a SYNC character has been received, the receiver will discard this character if the valid bit is set.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
V	0	0	0	0	0	0	0	SYNC							

NOTE

When using 7-bit characters with parity, the parity bit should be included in the SYNC register value.

7.10.20.8 BDLE-BISYNC DLE REGISTER. The 16-bit, memory-mapped, read-write BDLE register is used to define the BISYNC stripping and insertion of the DLE character. When the BISYNC controller is in transparent mode and an underrun occurs during message transmission, the BISYNC controller inserts DLE-SYNC pairs until the next data buffer is available for transmission.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
V	0	0	0	0	0	0	0	DLE							

When the BISYNC receiver is in transparent mode and a DLE character is received, the receiver discards this character and excludes it from the BCS if the valid bit is set. If the second (next) character is a SYNC character, the BISYNC controller discards it and excludes it from the BCS. If the second character is a DLE, the BISYNC controller will write it to the buffer and include it in the BCS. If the character is not a DLE or SYNC, the BISYNC controller will examine the control characters table and act accordingly. If the character is not in the table, the buffer will be closed with the DLE follow character error (DLE) bit set. If the valid bit is not set, the receiver will treat the character as a normal character.

NOTE

When using 7-bit characters with parity, the parity bit should be included in the DLE register value.

7.10.20.9 TRANSMITTING AND RECEIVING THE SYNCHRONIZATION SEQUENCE. The BISYNC channel can be programmed to transmit and receive a synchronization pattern. The pattern is defined in the DSR. The length of the SYNC pattern is defined in the SYNL bits in the GSMR. The receiver synchronizes on the synchronization pattern that is located in the DSR. If the SYNL bits specify a non-zero synchronization pattern, then the transmitter sends the entire contents of the DSR prior to each frame, starting with the LSB first. Thus, the user may wish to repeat the desired SYNC pattern in the other DSR bits as well.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
4-BIT SYNC															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8-BIT SYNC															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-BIT SYNC															

7.11.7.15 SMC UART MASK REGISTER (SMCM). The SMCM is referred to as the SMC UART mask register when the SMC is operating as a UART. It is an 8-bit read-write register with the same bit format as the SMC UART event register. If a bit in the SMC UART mask register is a one, the corresponding interrupt in the event register will be enabled. If the bit is zero, the corresponding interrupt in the event register will be masked. This register is cleared upon reset.

7.11.8 SMC UART Example

The following list is an initialization sequence for 9600 baud, 8 data bits, no parity, and 1 stop bit operation of an SMC UART assuming a 25-MHz system frequency. BRG1 and SMC1 are used.

1. The SDCR (SDMA configuration register) should be initialized to $0740, rather than being left at its default value of $0000.
2. Configure the port B pins to enable the SMTXD1 and SMRXD1. Write PBPAR bits 6 and 7 with ones. Write PBDIR bits 6 and 7 with zeros. Write PBODR bits 6 and 7 with zeros.
3. Configure the BRG1. Write BRGC1 with $010144. The DIV16 bit is not used, and the divider is 162 (decimal). The resulting BRG1 clock is 16x the desired bit rate of the UART.
4. Connect the BRG1 clock to SMC1 using the SI. Write the SMC1 bit in SIMODE with a 0. Write the SMC1CS bits in SIMODE with 000.
5. Write RBASE and TBASE in the SMC parameter RAM to point to the Rx BD and Tx BD in the dual-port RAM. Assuming one Rx BD at the beginning of dual-port RAM and one Tx BD following that Rx BD, write RBASE with $0000 and TBASE with $0008.
6. Program the CR to execute the INIT RX & TX PARAMS command for this channel. For instance, to execute this command for SCC1, write $0001 to the CR. This command causes the RBPTR and TBPTR parameters of the serial channel to be updated with the new values just programmed into RBASE and TBASE.
7. Write RFCR with $18 and TFCR with $18 for normal operation.
8. Write MRBLR with the maximum number of bytes per receive buffer. For this case, assume 16 bytes, so MRBLR = $0010.
9. Write MAX_IDL with $0000 in the SMC UART-specific parameter RAM to disable the MAX_IDL functionality for this example.
10. Clear BRKLN and BRKEC in the SMC UART-specific parameter RAM for the sake of clarity.
11. Set BRKCR to $0001, so that if a STOP TRANSMIT command is issued, one break character will be sent.
12. Initialize the Rx BD. Assume the Rx data buffer is at $00001000 in main memory. Write $B000 to Rx_BD_Status. Write $0000 to Rx_BD_Length (not required—done for instructional purposes only). Write $00001000 to Rx_BD_Pointer.
13. Initialize the Tx BD. Assume the Tx data buffer is at $00002000 in main memory

(PAPAR) bit is cleared. Each pin is configured as a dedicated on-chip peripheral pin if the corresponding PAPAR bit is set.

When the port a pin is configured as a general-purpose I/O pin, the signal direction for that pin is determined by the corresponding control bit in the port A data direction register (PADIR). The port A I/O pin is configured as an input if the corresponding PADIR bit is cleared; it is configured as an output if the corresponding PADIR bit is set. All PAPAR bits and PADIR bits are cleared on total system reset, configuring all port A pins as general-purpose input pins.

Table 7-19. Port A Pin Assignment

Signal	Pin Function			
	PAPAR = 0	PAPAR = 1		Input to On-Chip Peripherals
		PADIR = 0	PADIR = 1	
PA0	PORT A0	RXD1	RXD4[1]	GND
PA1	PORT A1	TXD1	TXD4[1]	—
PA2	PORT A2	RXD2	—	GND
PA3	PORT A3	TXD2	—	—
PA4	PORT A4	RXD3	L1TXDB	Undefined
PA5	PORT A5	TXD3	L1RXDB	GND
PA6	PORT A6	RXD4	L1TXDA	Undefined
PA7	PORT A7	TXD4	L1RXDA	L1RXDA = GND
PA8	PORT A8	CLK1/TIN1/L1RCLKA	BRGO1	CLK1/TIN1/L1RCLKA = BRGO1
PA9	PORT A9	CLK2	TOUT1	CLK2 = GND
PA10	PORT A10	CLK3/TIN2/L1TCLKA	BRGO2	CLK3/TIN2/L1TCLKA = BRGO2
PA11	PORT A11	CLK4	TOUT2	CLK4 = CLK8
PA12	PORT A12	CLK5/TIN3	BRGO3	CLK5/TIN3 = BRGO3
PA13	PORT A13	CLK6/L1RCLKB	TOUT3	CLK6/L1RCLKB = GND
PA14	PORT A14	CLK7/TIN4	BRGO4	CLK7/TIN4 = BRGO4
PA15	PORT A15	CLK8/L1TCLKB	TOUT4	CLK8/L1TCLKB = GND

NOTES:
1: Only available on REV C mask or later. NOT Available on REV A or B. And when PA6 or PA7 is not used as TXD4 or RXD4 functions
Rev A mask is C63T
Rev B mask are C69T, and F35G
Current Rev C mask are E63C, E68C and F15W

If a port A pin is selected as a general-purpose I/O pin, it may be accessed through the port A data register (PADAT). Data written to the PADAT is stored in an output latch. If a port A pin is configured as an output, the output latch data is gated onto the port pin. In this case, when PADAT is read, the port pin itself is read. If a port A pin is configured as an input, data written to PADAT is still stored in the output latch but is prevented from reaching the port pin. In this case, when PADAT is read, the state of the port pin is read.

If an input to a peripheral is not supplied from a pin, then a default value is supplied to the on-chip peripheral as listed in Table 7-19.

Table 7-23. Encoding the Interrupt Vector

Interrupt Number	Interrupt Source Description	Lower 5 Bits of Vector
1F	Parallel I/O—PC0	11111
1E	SCC1	11110
1D	SCC2	11101
1C	SCC3	11100
1B	SCC4	11011
1A	Parallel I/O—PC1	11010
19	Timer 1	11001
18	Parallel I/O—PC2	11000
17	Parallel I/O—PC3	10111
16	SDMA Channel Bus Error	10110
15	IDMA1	10101
14	IDMA2	10100
13	Reserved	10011
12	Timer 2	10010
11	RISC Timer Table	10001
10	Reserved	10000
F	Parallel I/O—PC4	01111
E	Parallel I/O—PC5	01110
D	Reserved	01101
C	Timer 3	01100
B	Parallel I/O—PC6	01011
A	Parallel I/O—PC7	01010
9	Parallel I/O—PC8	01001
8	Reserved	01000
7	Timer 4	00111
6	Parallel I/O—PC9	00110
5	SPI	00101
4	SMC1	00100
3	SMC2 / PIP	00011
2	Parallel I/O—PC10	00010
1	Parallel I/O—PC11	00001
0	Error	00000

vector exists as the last entry in this table. The error vector is issued by the CPM if an interrupt was requested by the CPM but was masked by the user prior to being serviced by CPU32+ core and if no other pending interrupts for the CPM are present. The user should provide an error interrupt service routine, even if it is simply an RTE instruction.

9.1.1.2 CLOCKING STRATEGY. In this application, the system clock is generated from a 32.768-kHz crystal into the QUICC. The QUICC's internal phase-locked loop (PLL) then multiplies the frequency up to 25 MHz, and outputs 25 MHz on CLKO1 and 50 MHz on CLKO2. Neither CLKO pin is required for the application. It is recommended that the CLKO outputs be disabled in software to save power.

The use of a 32.768-kHz crystal is not a requirement in the application. A 4-MHz crystal or a 25-MHz external oscillator could have been used, if desired.

The QUICC clocking section allows for the clock oscillator to be kept running through the VDDSYN pin in a power-down situation. This section does not address low-power issues, however.

9.1.1.3 RESETTING THE QUICC. If a QUICC is configured to provide the global chip select, it will also provide an internal power-on reset generation. Thus, the reset support function is very simple. If a pushbutton switch is needed, it can be connected by an open-drain buffer to the hard reset ($\overline{\text{RESETH}}$) pin, once debounced. The soft reset ($\overline{\text{RESETS}}$) pin is not used in this design except to indicate that an internal QUICC soft reset is in progress.

9.1.1.4 INTERRUPTS. External interrupts may be brought into the QUICC through either the $\overline{\text{IRQx}}$ pins or parallel I/O pins. This design shows no external interrupts (the $\overline{\text{IRQ7}}$–$\overline{\text{IRQ1}}$ pins are pulled high), but this could be easily changed if desired. Without any external interrupts requiring autovector capability, the $\overline{\text{AVEC}}$ pin is also pulled high.

Internal interrupts from the QUICC may be generated in the SIM60 or the CPM. No additional hardware is required.

9.1.1.5 BUS ARBITRATION. This design assumes that no alternate bus masters exist in the system. Thus, $\overline{\text{BR}}$ is pulled high, and $\overline{\text{BGACK}}$ is not connected, but pulled high since it is an open-drain signal.

9.1.1.6 BREAKPOINT GENERATION. The QUICC can be used to generate a hardware breakpoint signal. The result of a breakpoint (either internally generated using the breakpoint address register or externally generated using the $\overline{\text{BKPT}}$ pin) is a CPU32+ breakpoint cycle. In this application, the $\overline{\text{BKPT}}$ pin is tied high and is not used.

9.1.1.7 BUS MONITOR FUNCTION. The QUICC can be programmed to monitor the bus for bus cycles that are not properly terminated. If $\overline{\text{AS}}$ is asserted but not negated, the cycle will terminate with the $\overline{\text{BERR}}$ pin being asserted.

9.1.1.8 SPURIOUS INTERRUPT MONITOR. The QUICC will watch for spurious interrupt cycles on the levels that it supports internally. If such a condition occurs, $\overline{\text{BERR}}$ will be asserted by the QUICC.

9.1.1.9 SOFTWARE WATCHDOG. If desired, the QUICC software watchdog can be used to generate a level 7 interrupt or a system reset. In this application, the software watchdog is configured in software to generate a reset. No additional hardware is required.

dled differently. See 9.4.2 Memory Interfaces.) To provide dynamic bus sizing, the MC68150 device may be added to the design shown here.

Parity is supported for both SRAM and DRAM arrays using the four-byte parity lines PRTY3–PRTY0. When a parity error occurs, the error indication on the $\overline{\text{PERR}}$ pin causes the QUICC to generate a level 5 interrupt to the MC68EC040. (Level 7 has already been used for the breakpoint generation interrupt.) The parity error timing is not fast enough to allow an MC68EC040 bus error to be generated on the bus cycle that generated a parity error.

The QUICC supports MC68EC040 bursting using the BADD3–BADD2 pins. These pins normally reflect the values on A3–A2, but, in the case of a burst, are used to increment the address to the memory array. If the memory devices already support MC68EC040 bursting internally, the BADD3–BADD2 pins are not required.

The DRAM arrays require the four $\overline{\text{CAS3}}$–$\overline{\text{CAS0}}$ pins. Also, since an external address multiplexer is used, the AMUX pin is required to select between rows and columns. If, however, the user's configuration does not require DRAM, the AMUX pin can be used as an $\overline{\text{OE}}$ pin instead. This would save an inverter in a number of memory arrays.

NOTE

Many memory arrays show an inverter on the R/$\overline{\text{W}}$ pin to create the $\overline{\text{OE}}$ signal. When using multiple memory arrays, it is possible to share one inverter between multiple memory arrays; however, this configuration is not shown.

The QUICC also provides four write enable ($\overline{\text{WEx}}$) pins to select the correct byte during write operations.

9.4.2.2 REGULAR EPROM. Figure 9-9 shows the glueless interface to standard EPROM in the system. The assumption is made that only the MC68EC040 will access this array. No bursting capability is used. The CONFIG2–CONFIG0 pins are configured to initialize the system to slave mode, $\overline{\text{CS0}}$ operating on a 32-bit port at reset, the MBAR at its normal location, and MC68040 companion mode.

It would have been possible to use 16-bit-wide EPROM to reduce the chip count, if desired. (See 9.4.2.3 Burst EPROM. for an example.)

9.4.2.3 BURST EPROM. Figure 9-10 shows the glueless interface to two burst EPROMs available from National Semiconductor. These devices support a glueless interface to the MC68040. In this design, the assumption is made that only the MC68EC040 will access this array.

9.6.7.2 CONFIGURING THE MEMORY CONTROLLER. The following paragraphs describe configuring the memory controller.

For information on configuring the global memory register (GMR), refer to 9.1 Minimum System Configuration.

The memory controller status register (MSTAT) is used for reporting parity errors and does not require initialization.

Eight base registers (BRs) exist, one for each memory bank. BR6 and BR7 for $\overline{CS6}$ and $\overline{CS7}$ will be specified with the 53C90 address being $04001000. Please refer to 9.1 Minimum System Configuration for DRAM and other memory configurations.

BR7 = $0400104D. Address decoded is $04001xxx, function codes are xxxx (don't care, will be masked in OR7), TRLXQ and CSNTQ are set, parity is enabled, read and write accesses are allowed, and this base register is valid.

BR6 = $04001805. Same as BR7, except TRLXQ and CSNTQ are not set and the next consecutive 2K memory block is selected.

Eight option registers (ORs) exist, one for each memory bank. The following information is valid for registers OR6 and OR7:

DSSEL should be 0.

SPS1–SPS0 should be 10 (indicating port size is 8 bits).

PGME should be 0 since this is not DRAM.

BCYC1–BCYC0 are not used and should be cleared.

FCM3–FCM0 may be cleared to zeros to allow the chip select or $\overline{RAS}$ line to assert on all function codes, except CPU space (interrupt acknowledge). It is advisable to program FCM3–FCM0 to zeros, at least during the initial stages of debugging.

The AM27–AM11 bits will mask the address if they are cleared. In this application, they are all set to allow decoding.

The TCYC bits should be set to determine the number of wait states required—one wait state on $\overline{CS7}$ (0010) and no wait states on $\overline{CS6}$ (0001).

Therefore, OR7 = $2FFFF804 and OR6 = $1FFFF804.

9.7 USING THE QUICC AS A TAP CONTROLLER FOR BOARD SELF-TEST

An assembled board is often tested with complex test equipment using a unique test port or a bed-of-nails fixture. This procedure becomes more difficult as device packages and features become smaller. The objective of the JTAG standard is to define a boundary scan architecture that can be adopted as a part of an integrated circuit to perform both an in-circuit test and a verification of the interconnection between different devices.

The JTAG standard defines test logic that can be integrated into a device to perform:

1. Testing of the interconnection between devices once they have been mounted on a printed circuit board or any other substrate.

10.28 SPI MASTER ELECTRICAL SPECIFICATIONS

(The electrical specifications in this document are preliminary. See Figure 10-75 and Figure 10-76)

Num.	Characteristic	3.3 V or 5.0 V		5.0 V		Unit
		25.0 MHz		33.34 MHz		
		Min	Max	Min	Max	
160	Master Cycle Time	4	1024	4	1024	tcyc
161	Master Clock (SPICLK) High or Low Time	2	512	2	512	tcyc
162	Master Data Setup Time (Inputs)	50	—	50	—	ns
163	Master Data Hold Time (Inputs)	0	—	0	—	ns
164	Master Data Valid (after SPICLK Edge)	—	20	—	20	ns
165	Master Data Hold Time (Outputs)	0	—	0	—	ns
166	Rise Time: Output		15		15	s
167	Fall Time: Output		15		15	ns

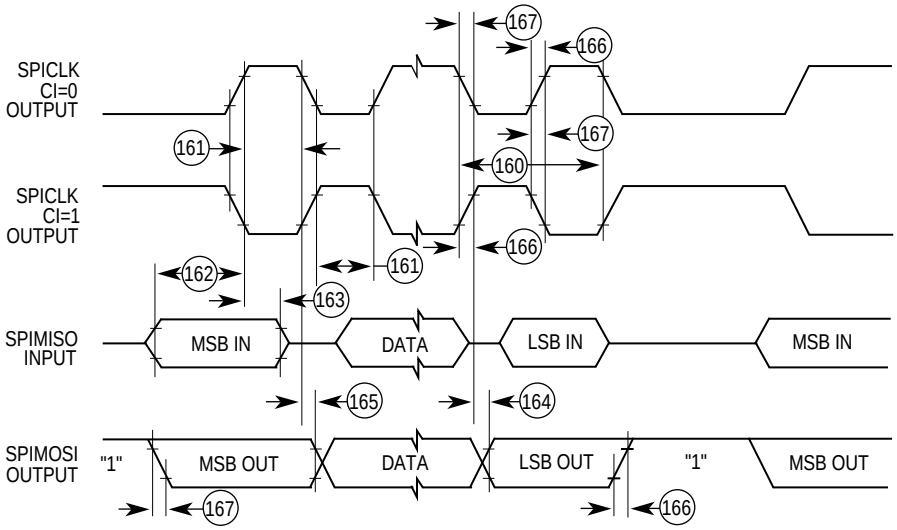

Figure 10-75. SPI Master (CP = 0)

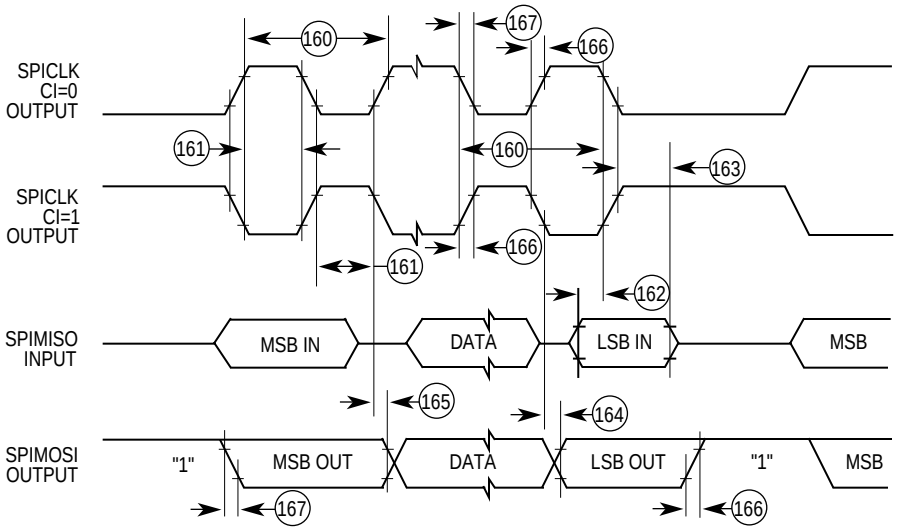

Figure 10-76. SPI Master (CP = 1)

- End Delimiter (ED) generation/checking
- Init Rx/Tx, Stop TX, Restart TX and Enter Hunt Mode Commands
- Token Rotation Timer, Idle Timer, Slot Timer and Time-out Timer
- Consumes 1280 bytes of the QUICC's internal memory.

C.6 ENHANCED ETHERNET FILTERING

The enhanced Ethernet filtering microcode has been created to add a bit more flexibility to the current Ethernet Controller. It allows the user to perfectly accept or reject frames with destination addresses that are contained in a table of 24 entries. In addition to the filtering capability, the microcode adds the ability to replace the externally sampled tag byte with one that is extracted from the address table.

C.6.1 Key Features

- Can accept or reject incoming frames if the destination MAC address is contained in a list of 24 preprogrammed entries.
- Allows a shorter list to obtain better peripheral performance.
- Can optionally tag incoming accepted frames with an 8-bit tag appended to the end of the frame (rather than using a CAM).
- The filtering can be turned off to allow tagging with no frame rejection.
- Can filter on group and individual addresses.
- Is a "small" microcode (consumes 768 bytes of DPRAM).
- Filtering is only supported on one Ethernet channel (SCC1), the other channel operates normally.

C.6.2 Performance

The overhead incurred by the filtering algorithm will depend upon the number of entries in the address table. Table C-5 shows possible configurations of other channels given the load added by a number of entries in the address table.

Table C-5. Channel Configuration

Number of addresses	Possible Configuration of Other Channels
1-8	1 x 10 Mbit Ethernet, 1 x 200 Kbit HDLC
9-16	1 x 10 Mbit Ethernet, 1 x 200 Kbit HDLC
17-24	1 x 10 Mbit Ethernet