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	-40°C ~ 85°C (TA)
Security Features	-
Package / Case	240-BFQFP
Supplier Device Package	240-FQFP (32x32)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/kmc68360cai25l

Table 3-2. CPM Sub-Module Base Addresses

3	SMC1 Base	DPRBASE + $E80
4	SCC4 Base	DPRBASE + $F00
4	IDMA2 Base	DPRBASE + $F70
4	SMC2 Base	DPRBASE + $F80

3.3 INTERNAL REGISTERS MEMORY MAP

In addition to the internal dual-port RAM, there are a number of internal registers to support the functions of the various CPU32+ core peripherals. The internal registers (see Table 3-3 and Table 3-4) are memory-mapped registers offset from the register base (REGBASE) pointer. REGBASE (abbreviated REGB) = DPRBASE + 4K. All registers are located on the internal IMB.

NOTES

All registers that are underlined in the following tables are special registers called event registers. In these registers, bits are set by the QUICC and cleared by the user. To clear a bit, the user must write a one to that bit. For example, to clear bit 2 in SCCE1, the MOVE.B #$04,SCCE1 instruction may be used. Do NOT use read-modify-write instructions (such as BSET, BCLR, AND, OR, etc.) with these registers, or ALL bits in that register will inadvertently be cleared. See the individual register descriptions for more information.

All undefined and reserved bits within registers and parameter RAM values written by the user should be written with zero to allow for future enhancements to the device.

Bold letters mark registers that are restricted to supervisor access.

3.3.1 SIM Registers Memory Map

Table 3-3 lists the SIM registers memory map.

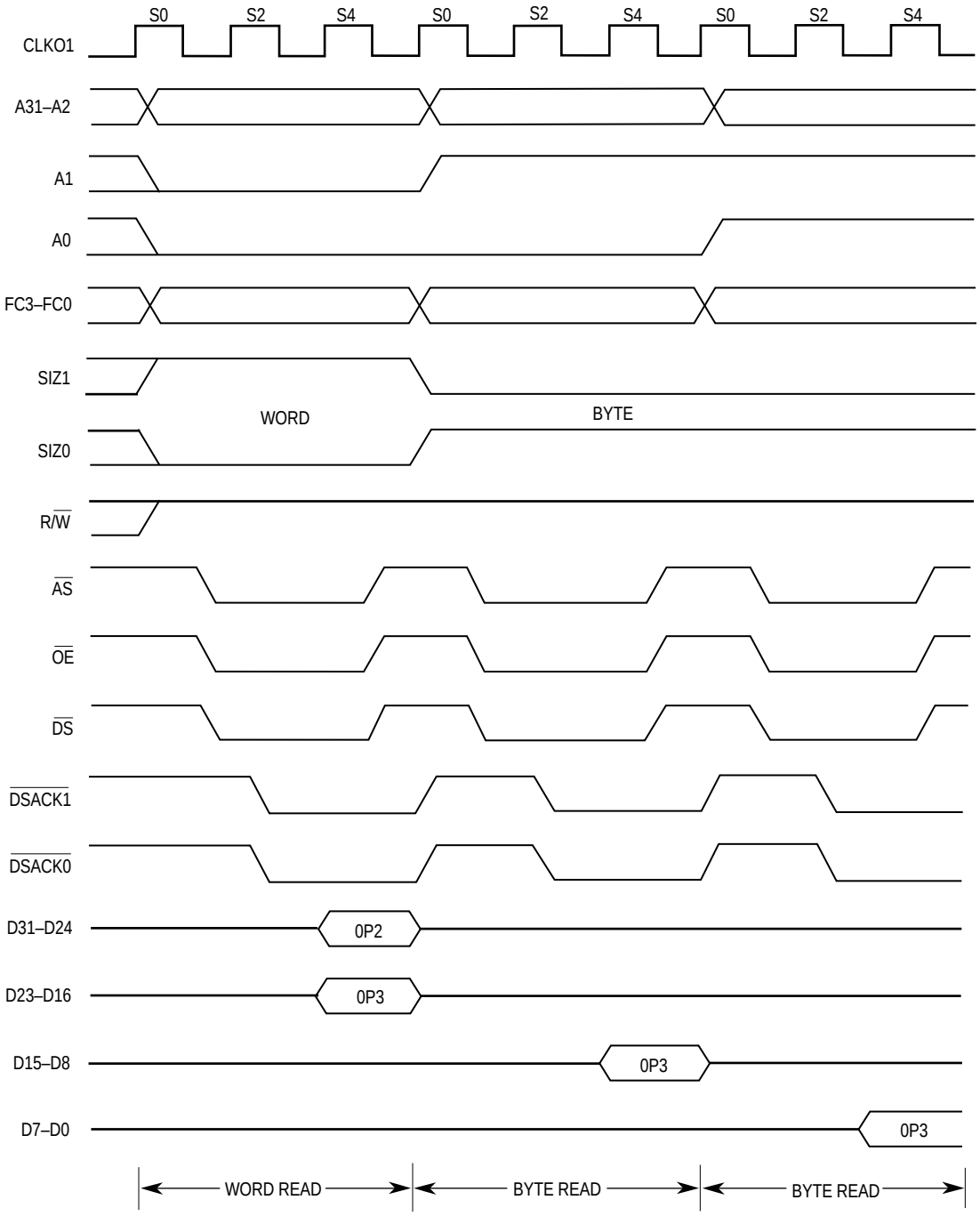


Figure 4-17. Byte and Word Read Cycles—32-Bit Port Timing

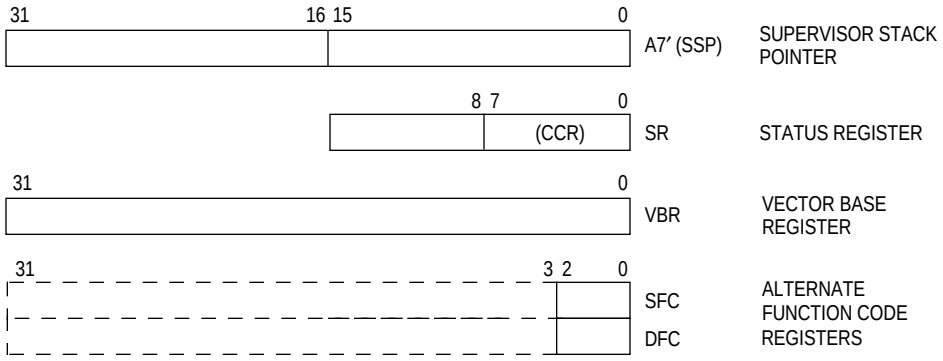


Figure 5-4. Supervisor Programming Model Supplement

5.2.2 Registers

Registers D7–D0 are used as data registers for bit, byte (8-bit), word (16-bit), long-word (32-bit), and quad-word (64-bit) operations. Registers A6 to A0 and the USP and SSP are address registers that may be used as software SPs or base address registers. Register A7 (shown as A7 and A7' in Figure 5-3 and Figure 5-4) is a register designation that applies to the USP in the user privilege level and to the SSP in the supervisor privilege level. In addition, address registers may be used for word and long-word operations. All 16 general-purpose registers (D7–D0, A7–A0) may be used as index registers.

The Program Counter (PC) contains the address of the next instruction to be executed by the CPU32+. During instruction execution and exception processing, the processor automatically increments the contents of the PC or places a new value in the PC, as appropriate.

The Status Register (SR) (see Figure 5-5) contains condition codes, an interrupt priority mask (three bits), and three control bits. Condition codes reflect the results of a previous operation. The codes are contained in the low byte (CCR) of the SR. The interrupt priority mask determines the level of priority an interrupt must have to be acknowledged. The control bits determine trace mode and privilege level. At user privilege level, only the CCR is available. At supervisor privilege level, software can access the full SR.

The Vector Base Register (VBR) contains the base address of the exception vector table in memory. The displacement of an exception vector is added to the value in this register to access the vector table.

Alternate source and destination function code registers (SFC and DFC) contain 3-bit function codes. The CPU32+ generates a function code each time it accesses an address. Specific codes are assigned to each type of access. The codes can be used to select eight dedicated 4-Gbyte address spaces. The MOVEC instruction can use registers SFC and DFC to specify the function code of a memory address.

Table 5-2. Instruction Set Summary (Concluded)

Opcode	Operation	Syntax
RTS	(SP) ⇒ PC; SP + 4 ⇒ SP	RTS
SBCD	$Destination_{10}$ – $Source_{10}$ – X ⇒ Destination	SBCD Dx,Dy SBCD –(Ax),–(Ay)
Scc	If Condition True then 1s ⇒ Destination else 0s ⇒ Destination	Scc ⟨ea⟩
STOP	If supervisor state then Immediate Data ⇒ SR; STOP else TRAP	STOP #⟨data⟩
SUB	Destination – Source ⇒ Destination	SUB ⟨ea⟩,Dn SUB Dn,⟨ea⟩
SUBA	Destination – Source ⇒ Destination	SUBA ⟨ea⟩,An
SUBI	Destination – Immediate Data ⇒ Destination	SUBI #⟨data⟩,⟨ea⟩
SUBQ	Destination – Immediate Data ⇒ Destination	SUBQ #⟨data⟩,⟨ea⟩
SUBX	Destination – Source – X ⇒ Destination	SUBX Dx,Dy SUBX –(Ax),–(Ay)
SWAP	Register [31:16] ⇔ Register [15:0]	SWAP Dn
TAS	Destination Tested ⇒ Condition Codes; 1 ⇒ bit 7 of Destination	TAS ⟨ea⟩
TBLS	ENTRY(n) + {(ENTRY(n + 1) – ENTRY(n)) × Dx[7:0]} / 256 ⇒ Dx	TBLS.⟨size⟩ ⟨ea⟩, Dx TBLS.⟨size⟩ Dym:Dyn, Dx
TBLSN	ENTRY(n) × 256 + {(ENTRY(n + 1) – ENTRY(n)) × Dx [7:0]} ⇒ Dx	TBLSN.⟨size⟩ ⟨ea⟩,Dx TBLSN.⟨size⟩ Dym:Dyn, Dx
TBLU	ENTRY(n) + {(ENTRY(n + 1) – ENTRY(n)) × Dx[7:0]} / 256 ⇒ Dx	TBLU.⟨size⟩ ⟨ea⟩,Dx TBLU.⟨size⟩ Dym:Dyn, Dx
TBLUN	ENTRY(n) × 256 + {(ENTRY(n + 1) – ENTRY(n)) × Dx[7:0]} ⇒ Dx	TBLUN.⟨size⟩ ⟨ea⟩,Dx TBLUN.⟨size⟩ Dym:Dyn,Dx
TRAP	SSP – 2 ⇒ SSP; Format/Offset ⇒ (SSP); SSP – 4 ⇒ SSP; PC ⇒ (SSP); SSP – 2 ⇒ SSP; SR ⇒ (SSP); Vector Address ⇒ PC	TRAP #⟨vector⟩
TRAPcc	If cc then TRAP	TRAPcc TRAPcc.W #⟨data⟩ TRAPcc.L #⟨data⟩
TRAPV	If V then TRAP	TRAPV
TST	Destination Tested ⇒ Condition Codes	TST ⟨ea⟩
UNLK	An ⇒ SP; (SP) ⇒ An; SP + 4 ⇒ SP	UNLK An

NOTE 1: d is direction, L or R.

5.3.3.1 CONDITION CODE REGISTER. The CCR portion of the SR contains five bits that indicate the result of a processor operation. Table 5-2 lists the effect of each instruction on these bits. The carry bit and the multiprecision extend bit are separate in the M68000 Family to simplify programming techniques that use them. Refer to Table 5-3 as an example.

Table 5-6. Logic Operations

Instruction	Operand Syntax	Operand Size	Operation
AND	⟨ea⟩, Dn Dn, ⟨ea⟩	8, 16, 32 8, 16, 32	Source Λ Destination ⇒ Destination
ANDI	#⟨data⟩, ⟨ea⟩	8, 16, 32	Immediate Data Λ Destination ⇒ Destination
EOR	Dn, ⟨ea⟩	8, 16, 32	Source ⊕ Destination ⇒ Destination
EORI	#⟨data⟩, ⟨ea⟩	8, 16, 32	Immediate Data ⊕ Destination ⇒ Destination
NOT	⟨ea⟩	8, 16, 32	$\overline{\text{Destination}}$ ⇒ Destination
OR	⟨ea⟩, Dn Dn, ⟨ea⟩	8, 16, 32 8, 16, 32	Source V Destination ⇒ Destination
ORI	#⟨data⟩, ⟨ea⟩	8, 16, 32	Immediate Data V Destination ⇒ Destination
TST	⟨ea⟩	8, 16, 32	Source – 0, to set condition codes

5.3.3.5 SHIFT AND ROTATE INSTRUCTIONS. The arithmetic shift instructions, ASR and ASL, and logical shift instructions, LSR and LSL, provide shift operations in both directions. The ROR, ROL, ROXR, and ROXL instructions perform rotate (circular shift) operations, with and without the extend bit. All shift and rotate operations can be performed on either registers or memory.

Register shift and rotate operations shift all operand sizes. The shift count may be specified in the instruction operation word (to shift from 1 to 8 places) or in a register (modulo 64 shift count).

Memory shift and rotate operations shift word-length operands one bit position only. The SWAP instruction exchanges the 16-bit halves of a register. Performance of shift/rotate instructions is enhanced so that use of the ROR and ROL instructions with a shift count of eight allows fast byte swapping. Table 5-7 is a summary of the shift and rotate operations.

Table 5-8. Bit Manipulation Operations

Instruction	Operand Syntax	Operand Size	Operation
BCHG	Dn, ⟨ea⟩ #⟨data⟩, ⟨ea⟩	8, 32 8, 32	~(⟨bit number⟩ of destination) ⇒ Z ⇒ bit of destination
BCLR	Dn, ⟨ea⟩ #⟨data⟩, ⟨ea⟩	8, 32 8, 32	~(⟨bit number⟩ of destination) ⇒ Z; 0 ⇒ bit of destination
BSET	Dn, ⟨ea⟩ #⟨data⟩, ⟨ea⟩	8, 32 8, 32	~(⟨bit number⟩ of destination) ⇒ Z; 1 ⇒ bit of destination
BTST	Dn, ⟨ea⟩ #⟨data⟩, ⟨ea⟩	8, 32 8, 32	~(⟨bit number⟩ of destination) ⇒ Z

5.3.3.7 BINARY-CODED DECIMAL (BCD) INSTRUCTIONS. Five instructions support operations on BCD numbers. The arithmetic operations on packed BCD numbers are add decimal with extend (ABCD), subtract decimal with extend (SBCD), and negate decimal with extend (NBCD). Table 5-9 is a summary of the BCD operations.

Table 5-9. Binary-Coded Decimal Operations

Instruction	Operand Syntax	Operand Size	Operation
ABCD	Dn, Dn – (An), – (An)	8 8	$Source_{10}$ + $Destination_{10}$ + X ⇒ Destination
NBCD	⟨ea⟩	8 8	0 – $Destination_{10}$ – X ⇒ Destination
SBCD	Dn, Dn – (An), – (An)	8 8	$Destination_{10}$ – $Source_{10}$ – X ⇒ Destination

5.3.3.8 PROGRAM CONTROL INSTRUCTIONS. A set of subroutine call and return instructions and conditional and unconditional branch instructions perform program control operations. Table 5-10 summarizes these instructions.

The vector number for the TRAP instruction is internally generated—part of the number comes from the instruction itself. The trap vector number, PC value, and a copy of the SR are saved on the supervisor stack. The saved PC value is the address of the instruction that follows the instruction that generated the trap. For all instruction traps other than TRAP, a pointer to the instruction causing the trap is also saved in the fifth and sixth words of the exception stack frame.

5.5.2.5 SOFTWARE BREAKPOINTS. To support hardware emulation, the CPU32+ must provide a means of inserting breakpoints into target code and of announcing when a breakpoint is reached.

The MC68000 and MC68008 can detect an illegal instruction inserted at a breakpoint when the processor fetches from the illegal instruction exception vector location. Since the VBR on the CPU32+ allows relocation of exception vectors, the exception vector address is not a reliable indication of a breakpoint. CPU32+ breakpoint support is provided by extending the function of a set of illegal instructions ($4848–$484F).

When a breakpoint instruction is executed, the CPU32+ performs a read from CPU space $0, at a location corresponding to the breakpoint number. If this bus cycle is terminated by $\overline{\text{BERR}}$, the processor performs illegal instruction exception processing. If the bus cycle is terminated by $\overline{\text{DSACK}}$x, the processor uses the data returned to replace the breakpoint in the instruction pipeline and begins execution of that instruction. See Section 4 Bus Operation for a description of CPU space operations.

5.5.2.6 HARDWARE BREAKPOINTS. The CPU32+ recognizes hardware breakpoint requests. Hardware breakpoint requests do not force immediate exception processing, but are left pending. An instruction breakpoint is not made pending until the instruction corresponding to the request is executed.

A pending breakpoint can be acknowledged between instructions or at the end of exception processing. To acknowledge a breakpoint, the CPU performs a read from CPU space $0 at location $1E (see Section 4 Bus Operation).

If the bus cycle terminates normally, instruction execution continues with the next instruction as if no breakpoint request occurred. If the bus cycle is terminated by $\overline{\text{BERR}}$, the CPU begins exception processing. Data returned during this bus cycle is ignored.

Exception processing follows the regular sequence. Vector number 12 (offset $30) is internally generated. The PC of the executing instruction, the PC of the next instruction to be executed, and a copy of the SR are saved on the supervisor stack.

5.5.2.7 FORMAT ERROR. The processor checks certain data values for control operations. The validity of the stack format code and, in the case of a bus cycle fault format, the version number of the processor that generated the frame are checked during execution of the RTE instruction. This check ensures that the program does not make erroneous assumptions about information in the stack frame.

If the format of the control data is improper, the processor generates a format error exception. This exception saves a four-word format exception frame and then vectors through vec-

tial general system clock of 13.14 MHz. The user would then write the MF bits to raise this frequency to the desired frequency.

NOTE

SWT clocking does not stop when the PLL is in the process of acquiring a lock. Therefore, the user should service the SWT (reset its count) before and after changing the MF bits.

6.9.3.11 CLOCK DIVIDER CONTROL REGISTER (CDVCR). The CDVCR controls the operation of the low-power divider for the various clocks on the QUICC. It can be read or written only in supervisor mode. Writing this register is allowed only if the CDVWP bit is zero. The reset state of CDVCR produces the maximum frequency for all the clocks that it affects.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDVWP	DFSY		DFTM		INTEN			RRQEN	DFNL			DFNH			CSRC
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CDVWP—CDVCR Write Protect

This bit protects accidental writing of the CDVCR. After reset, this bit defaults to zero to enable writing. Setting this bit prevents further writing (excluding the first write that sets this bit).

DFSY—Division Factor for the SyncCLK

These bits define the SyncCLK frequency. Changing the value of the these bits will not result in a loss-of-lock condition. These bits are cleared by a hardware reset. The default value is divide by 1 (VCO/2) which is 25 MHz in a 25-MHz system.

00 = Divide by 1 (normal operation)
01 = Divide by 4
10 = Divide by 16
11 = Divide by 64

DFTM—Division Factor for the BRGCLK

These bits define the BRGCLK frequency. Changing the value of the these bits will not result in a loss-of-lock condition. These bits are cleared by a hardware reset. The default value is divide by 1 (VCO/2) which is 25 MHz in a 25-MHz system.

00 = Divide by 1 (normal operation)
01 = Divide by 4
10 = Divide by 16
11 = Divide by 64

INTEN—Interrupt Enable

These bits specify if the general system clock returns to high frequency (defined by the DFNH bits) *while* the CPU32+ either has a pending interrupt or an interrupt routine in process, either of which has a level higher than INTEN2–INTEN0. To prevent interrupts from causing the general system clock to automatically switch to high frequency, write INTEN with 111.

6.12.4 DRAM Bank Parity

Parity can be configured for any DRAM bank. Parity is generated and checked on a per-byte basis using PRTY3–PRTY0 if the PAREN bit is set in the BR. The OPAR bit in the GMR determines the type of parity (odd or even), and the PBEE bit in the GMR determines if an internal master should generate an error as a result of a parity error. Any parity error activates the $\overline{\text{PERR}}$ pin until the associated PERx bit in the MSTAT is cleared.

NOTE

Asynchronous external masters do not have parity support.

Parity is not supported for bus cycles terminated with external assertion of $\overline{\text{DSACK}}$ or $\overline{\text{TA}}$.

6.12.5 Refresh Operation

The DRAM controller uses $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh cycles. The refresh cycles are timed using a dedicated refresh timer. In the $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ method, the DRAMs have an internal refresh row address counter, so row addresses need not be supplied by the DRAM controller. These DRAMs recognize the assertion of $\overline{\text{CAS}}$ before the assertion of $\overline{\text{RAS}}$ and perform the refresh using their internal refresh row address value.

Each time the refresh timer expires, the DRAMC performs a refresh cycle. At the first opportunity after acquiring bus mastership, the DRAM controller requests the bus with the highest bus arbitration priority level 6. In addition, it asserts the $\overline{\text{BCLRO}}$ signal to minimize the delay before the refresh cycle begins, assuming the external bus master recognizes this signal and clears itself off the bus. Once the DRAM controller obtains the bus, it performs a refresh bus cycle to the DRAM bank.

If more than one bank of DRAM exists in the system, the user should program the refresh controller to request the bus more often (N times as often, where N is the number of banks). For instance, typical DRAMs require a refresh every 15.6 µs. If 2 banks of DRAM exist in the system, the DRAM controller should be programmed to refresh every 7.8 µs. In the two bank case, the DRAM controller will alternate between the banks, using the $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ technique on each bank every 7.8 µs.

The DRAM controller will automatically stack up to seven refresh requests before receiving the bus mastership. Once it receives the bus, it will perform all stacked cycles (up to seven), as sequential, back-to-back refresh bus cycles.

Refresh cycles are executed only when the RFEN bit in the GMR is set. The refresh cycle length (three to six clocks) is programmed by the RCYC bits in the GMR. The time between refreshes is programmed in the RCNT bits in the GMR (see 6.13.1 Global Memory Register (GMR)).

NOTE

DRAM banks normally need eight read cycles and some delay time after a power-on reset. After enabling the DRAM bank, the

7.8.5.4 SI COMMAND REGISTER (SICMR). The 8-bit SICMR allows the user to dynamically program the SI RAM. For more information about dynamic programming, refer to 7.8.4.7 SI RAM Dynamic Changes

The contents of this register are valid only in the RAM division mode (RDM1–RDM0 bits in SIGMR equal 01 or 11). This register is cleared at reset.

7	6	5	4	3	2	1	0
CSRRa	CSRTa	CSRRb	CSRTb	—			

CSRRx—Change Shadow RAM for TDM A or B Receiver

When set, this bit will cause the SI receiver to replace the current route with the shadow RAM. The bit is set by the user and cleared by the SI.

0 = The receiver shadow RAM is not valid. The user can write into the shadow RAM to program a new routing.
1 = The receiver shadow RAM is valid. The SI will exchange between the RAMs and take the new receive routing from the receiver shadow RAM. This bit is cleared as soon as the switch has completed.

CSRTx—Change Shadow RAM for TDM A or B Transmitter

When set, this bit will cause the SI transmitter to replace the current route with the shadow RAM. The bit is set by the user and cleared by the SI.

0 = The transmitter shadow RAM is not valid. The user can write into the shadow RAM to program a new routing.
1 = The transmitter shadow RAM is valid. The SI will exchange between the RAMs and take the new transmitter routing from the receiver shadow RAM. This bit is cleared as soon as the switch has completed.

Bits 3–0—Reserved

These bits should be set to zero by the user.

7.8.5.5 SI STATUS REGISTER (SISTR). The 8-bit SISTR indicates to the user which part of the SI RAM is the current-route RAM. The value of this register is valid only when the corresponding bit in the SIGMR is clear. This register is cleared at reset.

CRORa—Current Route of TDMa Receiver

7	6	5	4	3	2	1	0
CRORa	CROTa	CRORb	CROTb	—			

0 = The current-route receiver RAM is in address:
0–63 when the SI supports one TDM (RDM = 01)
0–31 when the SI supports two TDMs (RDM = 11)
1 = The current route receiver RAM is in address:
64–127 when the SI supports one TDM (RDM = 01)
32–63 when the SI supports two TDMs (RDM = 11)

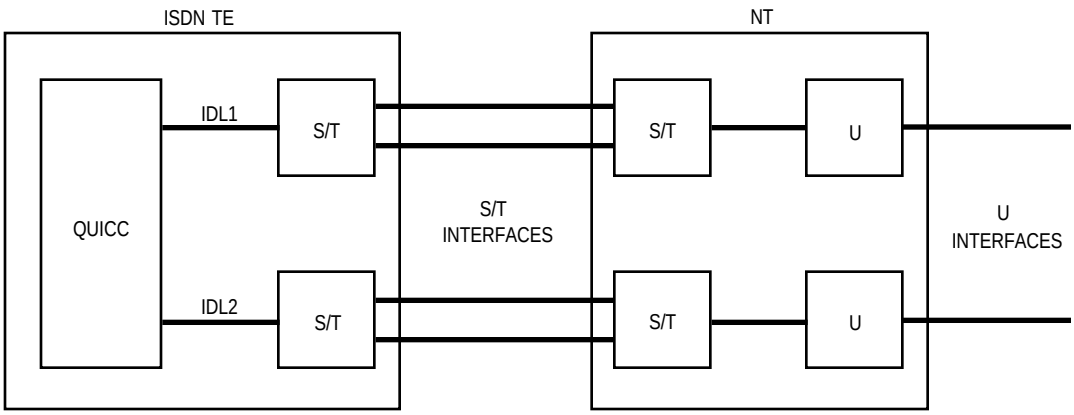


Figure 7-31. Dual IDL Bus Application Example

7.8.6.1 IDL INTERFACE EXAMPLE. An example of the IDL application is the ISDN terminal adaptor shown in Figure 7-32. In such an application, the IDL interface is used to connect the 2B+D channels between the QUICC, CODEC, and S/T transceiver. One of the QUICC SCCs would be configured to HDLC mode to handle the D channel; another QUICC SCC would be used to rate adapt the terminal data stream over the first B channel. That SCC would be configured for HDLC mode if V.120 rate adaption is required. The second B channel could be routed to the CODEC as a digital voice channel, if desired. The SPI is used to send initialization commands and periodically check status from the S/T transceiver. The SCC connected to the terminal would be configured for UART or other protocol depending on the terminal protocol used. Alternatively, instead of a terminal, a connection to a LAN could be made via Ethernet.

CRTx bits, and program the GRx bits to transfer the D channel grant to the SCC that supports this channel. The user should mark the received bit, which is the grant bit, by programming the channel select bits of the SI RAM to 111 for an internal assertion of a strobe on this bit. This bit will be sampled by the SI and transferred to the D channel SCC as the grant. The bit is generally bit 4 of the C/I in channel 2 of GCI, but any other bit may be selected using the SI RAM.

For example, assuming SCC1 is connected to the D channel, SCC2 is connected to the B1 channel, and SCC4 is connected to the B2 channel, SMC1 is used to handle the C/I channels, and the D channel grant is on bit 4 of the C/I on SCIT channel 2, the initialization sequence is as follows:

1. Program the SI RAM. Write all entries that are not used with $0001, setting the LST bit and disabling the routing function.

Entry No.	RAM Word						
	SWTR	SSEL	CSEL	CNT	BYT	LST	Description
1	0	0000	010	0000	1	0	8 Bits SCC2
2	0	0000	100	0000	1	0	8 Bits SCC4
3	0	0000	101	0000	1	0	8 Bits SMC1
4	0	0000	001	0001	0	0	2 Bits SCC1
5	0	0000	101	0101	0	0	6 Bits SMC1
6	0	0000	000	0110	1	0	Skip 7 Bytes
7	0	0000	000	0001	0	0	Skip 2 Bits
8	0	0000	111	0000	0	1	D Grant Bit

NOTE

Since GCI requires the same routing for both receive and transmit, an exact duplicate of the above entries should be written to both the receive and transmit sections of the SI RAM beginning at addresses 0 and 128, respectively.

2. SIMODE = $000080E0. Only TDMa is used; SMC1 is connected. SCIT mode is used in this example.

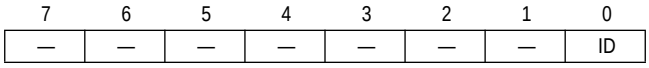
NOTE

If SCIT mode is not used, delete the last three entries of the SI RAM and set the LST bit in the new last entry.

3. SICR = $400040C0. SCC4, SCC2, and SCC1 are connected to the TSA. SCC1 supports the grant mechanism since it is on the D channel.
4. PAODR bit 6 = 1. Configures L1TXDa to an open-drain output.
5. PAPAR bits 6, 7, and 8 = 1. Configures L1TXDa, L1RXDa, and L1RCLKa.
6. PADIR bits 6 and 7 = 1. PADIR bit 8 = 0. Configures L1TXDa, L1RXDa, and L1RCLKa.

7.10.16.19 UART MASK REGISTER (SCCM). The SCCM is referred to as the UART mask register when the SCC is operating as a UART. It is a 16-bit read-write register with the same bit formats as the UART event register. If a bit in the 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.

7.10.16.20 SCC STATUS REGISTER (SCCS). The SCCs is an 8-bit read-only register that allows the user to monitor real-time status conditions on the RXD line. The real-time status of the $\overline{\text{CTS}}$ and $\overline{\text{CD}}$ pins are part of the port C parallel I/O.



Bits 7–1—Reserved

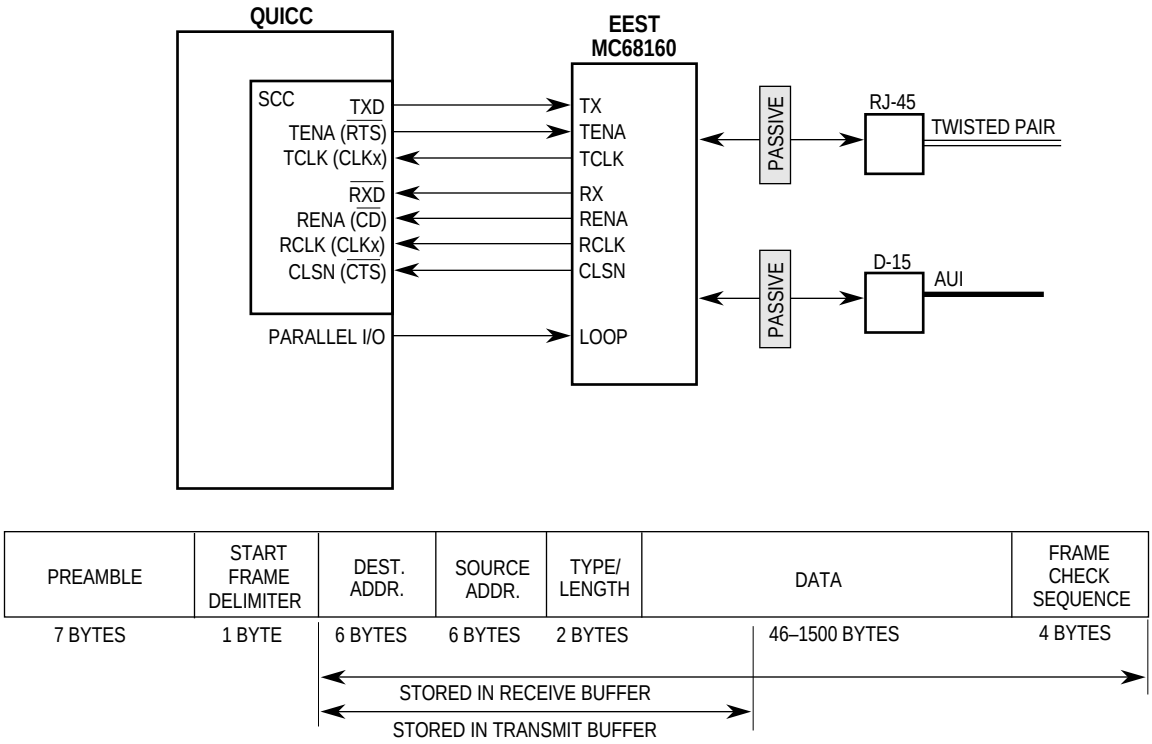
ID—Idle Status

ID is set when the RXD pin has been a logic one for at least one full character time.

0 = The line is not currently idle.
1 = The line is currently idle.

7.10.16.21 SCC UART EXAMPLE. The following list is an initialization sequence for 9600 baud, 8 data bits, no parity, and stop bit of an SCC UART operation assuming a 25-MHz system frequency. BRG1 and SCC4 are used. The UART is configured with the $\overline{\text{RTS4}}$, $\overline{\text{CTS4}}$, and $\overline{\text{CD4}}$ pins active. In addition, the $\overline{\text{CTS4}}$ pin is used as an automatic flow control signal.

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 A pins to enable the TXD4 and RXD4 pins. Write PAPAR bits 6 and 7 with ones. Write PADIR bits 6 and 7 with zeros. Write PAODR bits 6 and 7 with zeros.
3. Configure the port C pins to enable $\overline{\text{RTS4}}$, $\overline{\text{CTS4}}$, and $\overline{\text{CD4}}$. Write PCPAR bit 3 with one and bits 10 and 11 with zeros. Write PCDIR bits 3, 10, and 11 with zeros. Write PCSO bits 10 and 11 with ones.
4. Configure BRG1. Write BRGC1 with $010144. The DIV16 bit is not used, and the divider is 162 (decimal). The resulting BRG1 clock is 16× the desired bit rate of the UART.
5. Connect the BRG1 clock to SCC4 using the SI. Write the R4CS bits in SICR to 000. Write the T4CS bits in SICR to 000.
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 $0740 to the SDCR to initialize the SDMA Configuration Register.
8. Connect the SCC4 to the NMSI (i.e., its own set of pins). Clear the SC4 bit in the



NOTE: Short transmit frames are padded automatically by the QUICC.

Figure 7-67. Connecting the QUICC to Ethernet

The following pins take on new meanings when the Ethernet protocol is selected for the SCC:

1. Transmit Enable (TENA). The SCC's $\overline{\text{RTS}}$ pin changes to become TENA when the SCC is configured for Ethernet operation. The polarity of TENA is active high; whereas, the polarity of $\overline{\text{RTS}}$ is active low.
2. Receive Enable (RENA). The SCC's $\overline{\text{CD}}$ pin changes to become RENA when the SCC is configured for Ethernet operation. The polarity of RENA is active high; whereas, the polarity of $\overline{\text{CD}}$ is active low.
3. Collision (CLSN). The SCC's $\overline{\text{CTS}}$ pin changes to become CLSN when the SCC is configured for Ethernet operation. The polarity of CLSN is active high; whereas, the polarity of $\overline{\text{CTS}}$ is active low.

NOTE

The carrier sense signal is often referred to in Ethernet descriptions, because it defines whether the LAN is currently in use. Carrier sense is defined as RENA ORed with CLSN.

The EEST has similar names for its connection to the seven basic QUICC pins. In addition, the EEST contains a loopback pin to allow the QUICC to perform external loopback testing. This can be controlled by any available parallel I/O pin on the QUICC.

W—Wrap (Final BD in Table)

0 = This is not the last BD in the Rx BD table.
1 = This is the last BD in the Rx BD table. After this buffer has been used, the CP will receive incoming data into the first BD in the table (the BD pointed to by RBASE). The number of Rx BDs in this table is programmable and is determined only by the W-bit and the overall space constraints of the dual-port RAM.

I—Interrupt

0 = No interrupt is generated after this buffer has been filled.
1 = The RX bit in the event register will be set when this buffer has been completely filled by the CP, indicating the need for the CPU32+ core to process the buffer. The RX bit can cause an interrupt if it is enabled.

CM—Continuous Mode

0 = Normal operation.
1 = The E-bit is not cleared by the CP after this BD is closed, allowing the associated data buffer to be overwritten automatically when the CP next accesses this BD. However, the E-bit will be cleared if an error occurs during reception, regardless of the CM bit.

The following status bits are written by the CP after the received data has been into the associated data buffer.

ID—Buffer Closed on Reception of Idles

The buffer was closed due to the reception of the programmable number of consecutive idle sequences.

BR—Buffer Closed on Reception of Break

The buffer was closed due to the reception of a break sequence.

FR—Framing Error

A character with a framing error was received and is located in the last byte of this buffer. A framing error is a character without a stop bit. A new receive buffer will be used for further data reception.

PR—Parity Error

A character with a parity error was received and is located in the last byte of this buffer. A new receive buffer will be used for further data reception.

OV—Overrun

A receiver overrun occurred during message reception.

Data Length

Data length is the number of octets that the CP has written into this BD's data buffer. It is written only once by the CP as the BD is closed.

CRXB—C/I Channel Buffer Received

The C/I receive buffer is full.

MTXB—Monitor Channel Buffer Transmitted

The monitor transmit buffer became empty.

MRXB—Monitor Channel Buffer Received

The monitor receive buffer is full.

7.11.14.10 SMC MASK REGISTER (SMCM). The SMCM is an 8-bit, memory-mapped, read-write register. It has the same bit format as the SMC event register. If a bit in the SMCM is a one, the corresponding interrupt in the SMC event register will be enabled. If the bit is zero, the corresponding interrupt in the SMC event register will be masked. The SMCM is clear upon reset.

7.12 SERIAL PERIPHERAL INTERFACE (SPI)

The SPI allows the QUICC to exchange data between other QUICC chips, the MC68302, the M68HC11 and M68HC05 microcontroller families, and a number of peripheral devices such as EEPROMs, real-time clock devices, A/D converters, and ISDN devices.

7.12.1 Overview

The SPI is a full-duplex, synchronous, character-oriented channel that supports a four-wire interface (receive, transmit, clock, and slave select).

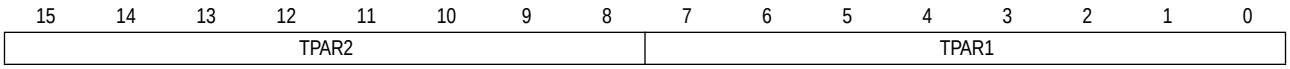
The SPI block consists of transmitter and receiver sections, an independent baud rate generator, and a control unit. The transmitter and receiver sections use the same clock, which is derived from the SPI baud rate generator in master mode and generated externally in slave mode. During an SPI transfer, data is transmitted and received simultaneously. Refer to Figure 7-80 for the SPI block diagram.

T/R—Transmit/Receive Select

This bit selects transmitter or receiver operation for the PIP when it is using the interlocked, pulsed, or transparent handshake modes.

0 = Data is input to the PIP.
1 = Data is output from the PIP.

7.13.7.3 PIP TIMING PARAMETERS REGISTER (PTPR). The PTPR is a 16-bit read-write register that is cleared at reset. The PTPR holds two timing parameters, TPAR1 and TPAR2, which are used in the pulsed handshake modes for both a PIP transmitter and a receiver.



TPAR1—Timing Parameter 1

This 8-bit value defines the number of system clocks for TPAR1 in the transmitter or receiver pulsed handshake mode. The value $00 corresponds to 1 QUICC general system clock, and the value $FF corresponds to 256 QUICC general system clocks. A general system clock defaults to 40 ns, assuming a 25-MHz QUICC system.

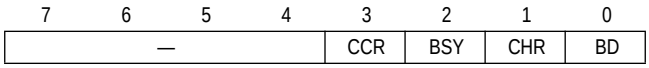
TPAR2—Timing Parameter 2

This 8-bit value defines the number of system clocks for TPAR2 in the transmitter or receiver pulsed handshake mode. The value $00 corresponds to 1 QUICC general system clock, and the value $FF corresponds to 256 QUICC general system clocks. A general system clock defaults to 40 ns, assuming a 25-MHz QUICC system.

7.13.7.4 PIP BUFFER DESCRIPTORS. BDs for the receiver and transmitter that support PIP operation were still in preparation at the time of writing.

7.13.7.5 PIP EVENT REGISTER (PIPE). The PIPE is an 8-bit register used to report events recognized by the PIP and to generate interrupts. It shares the same address as the SMC2 event register; thus, SMC2 cannot be used simultaneously with the PIP. Upon recognition of an event, the PIP sets its corresponding bit in the PIPE. Interrupts generated by this register may be masked in the PIP mask register.

The PIPE is a memory-mapped register that may be read at any time. A bit is cleared by writing a one (writing a zero does not affect a bit's value). More than one bit may be cleared at a time. All unmasked bits must be cleared before the CP will clear the internal interrupt request. This register is cleared at reset.



Bits 7–4—Reserved

CCR—Control Character Received

A control character was received (with reject (R) = 1) and stored in the receive control character register.

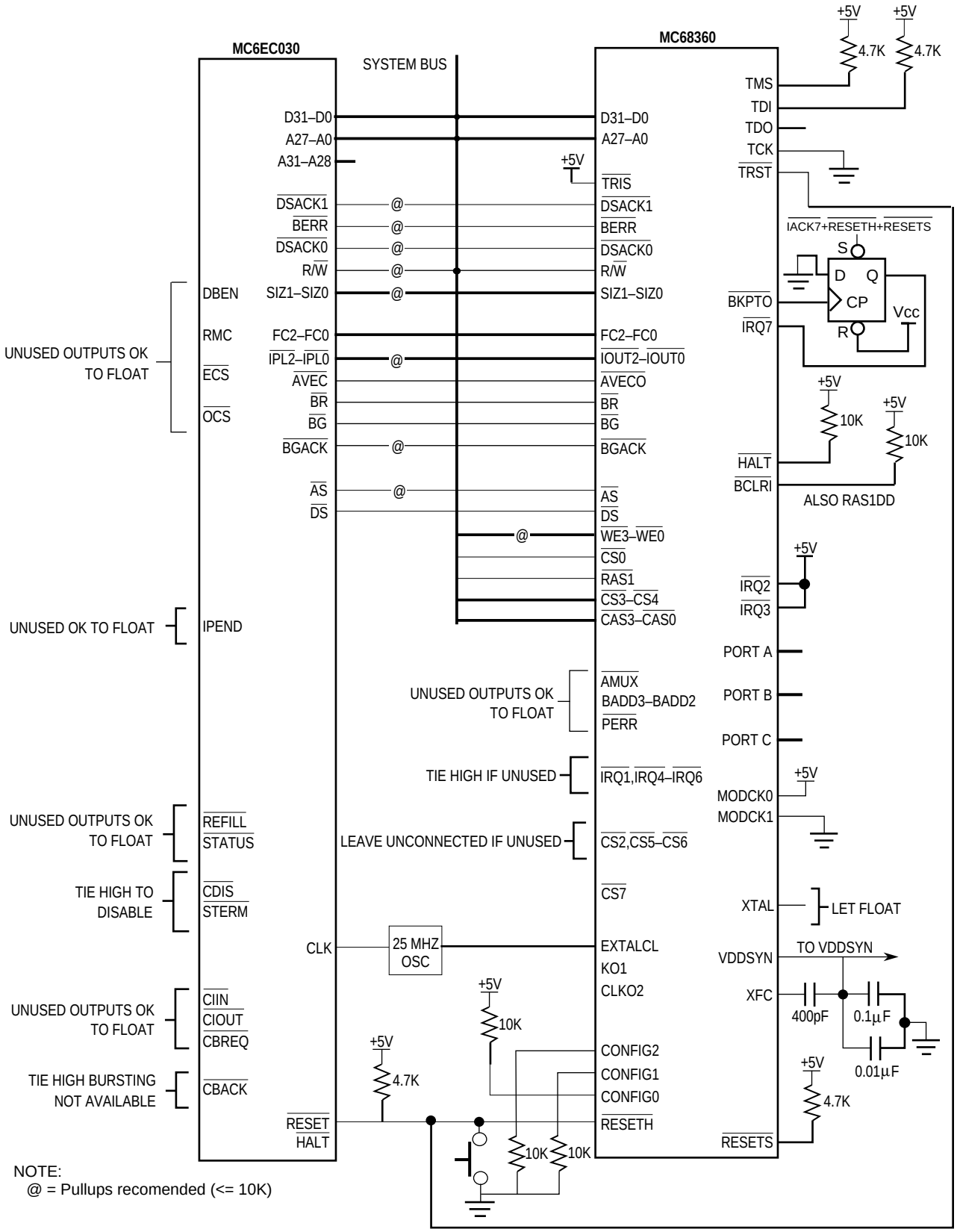


Figure 9-27. MC68EC030 to QUICC Interface

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

10.4 AC ELECTRICAL SPECIFICATION DEFINITIONS

The AC specifications presented consist of output delays, input setup and hold times, and signal skew times. All signals are specified relative to an appropriate edge of the clock and possibly to one or more other signals.

The measurement of the AC specifications is defined by the waveforms shown in Figure 10-1. To test the parameters guaranteed by Motorola, inputs must be driven to the voltage levels specified in the figure. Outputs are specified with minimum and/or maximum limits, as appropriate, and are measured as shown. Inputs are specified with minimum setup and hold times and are measured as shown. Finally, the measurement for signal-to-signal specifications are shown.

Note that the testing levels used to verify conformance to the AC specifications do not affect the guaranteed DC operation of the device as specified in the DC electrical characteristics.