

Philips Components

| Data sheet | |
|---------------|-----------------------|
| status | Product specification |
| date of issue | November 1990 |
| | |

SCC68070

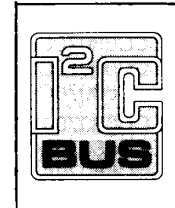
16/32-bit microprocessor

FEATURES

- CMOS technology
- 32-bit internal structure
- Enhanced bus error handling
- 4 decoded interrupt inputs
- 2 programmable interrupt inputs
- Decoded interrupt acknowledge
- Built-in clock generator - maximum 35 MHz crystal
- On-chip MMU; supporting virtual memory
- 2-channel DMA controller
- I²C serial bus interface
- UART serial bus interface
- 16-bit timer/counter
- Two 16-bit match/count/capture registers
- Fully 68000 object code compatible
- Bus interface similar to 68000
- 56 powerful instruction types
- 5 basic data types
- 16 Mbyte addressing range
- 14 addressing modes
- Memory mapped I/O
- Vectored and auto-vectored interrupts
- 7 interrupt levels
- Maximum internal clock frequency: 17.5 MHz
- 84-pin PLCC or a 120-pin QFP package

GENERAL DESCRIPTION

The SCC68070 is a 16/32-bit central processing unit suitable for use in a large variety of applications. It is fully object code compatible with the 68000. By integrating standard and advanced peripheral functions on the SCC68070, system costs are drastically reduced.



The internal architecture is built around a bus interconnecting the CPU and the various on-chip peripheral functions. Each function has several dedicated connections to the external circuitry. The SCC68070 includes powerful programmable interrupt processing circuitry for interrupts generated by internal and external sources. An on-chip clock generator provides a half crystal frequency clock signal for CPU and peripheral interfaces.

The on-chip MMU, if selected takes care of address translation and memory protection. Two DMA channels increase data throughput and the I²C-bus interface allows easy and low-cost addition of peripherals. The SCC68070 also includes a UART interface. A built-in timer/counter with two independently programmable match/count/capture registers, means that the SCC68070 can be programmed with two of the following options simultaneously:

- pulse generator
- external event counter
- reference timer

This document gives an overview of the basic functions, internal structure and electrical characteristics. For further details on the features and operation of the SCC68070 refer to "User Manual, Part 1 - Hardware".



16/32-bit microprocessor**SCC68070****ORDERING INFORMATION**

| EXTENDED TYPE NUMBER | PACKAGE | | | | CLOCK FREQUENCY (MHz) | TEMPERATURE RANGE (°C) |
|-------------------------|---------|--------------|----------|---------------|-----------------------------|---------------------------|
| | PINS | PIN POSITION | MATERIAL | CODE | | |
| SCC68070CBA84 | 84 | PLCC | plastic | SOT189CG, AGA | 12.5 | 0 to 70 |
| SCC68070CCA84 | 84 | PLCC | plastic | SOT189CG, AGA | 15.0 | 0 to 70 |
| SCC68070CDA84 | 84 | PLCC | plastic | SOT189CG, AGA | 17.5 | 0 to 70 |
| SCC68070ABA84 | 84 | PLCC | plastic | SOT189CG, AGA | 12.5 | -40 to 85 |
| SCC68070ACA84 | 84 | PLCC | plastic | SOT189CG, AGA | 15.0 | -40 to 85 |
| SCC68070ADA84 | 84 | PLCC | plastic | SOT189CG, AGA | 17.5 | -40 to 85 |
| SCC68070CBB | 120 | QFP | plastic | SOT220 | 12.5 | 0 to 70 |
| SCC68070CCB | 120 | QFP | plastic | SOT220 | 15.0 | 0 to 70 |
| SCC68070CDB | 120 | QFP | plastic | SOT220 | 17.5 | 0 to 70 |
| SCC68070ABB | 120 | QFP | plastic | SOT220 | 12.5 | -40 to 85 |
| SCC68070ACB | 120 | QFP | plastic | SOT220 | 15.0 | -40 to 85 |
| SCC68070ADB | 120 | QFP | plastic | SOT220 | 17.5 | -40 to 85 |

16/32-bit microprocessor

SCC68070

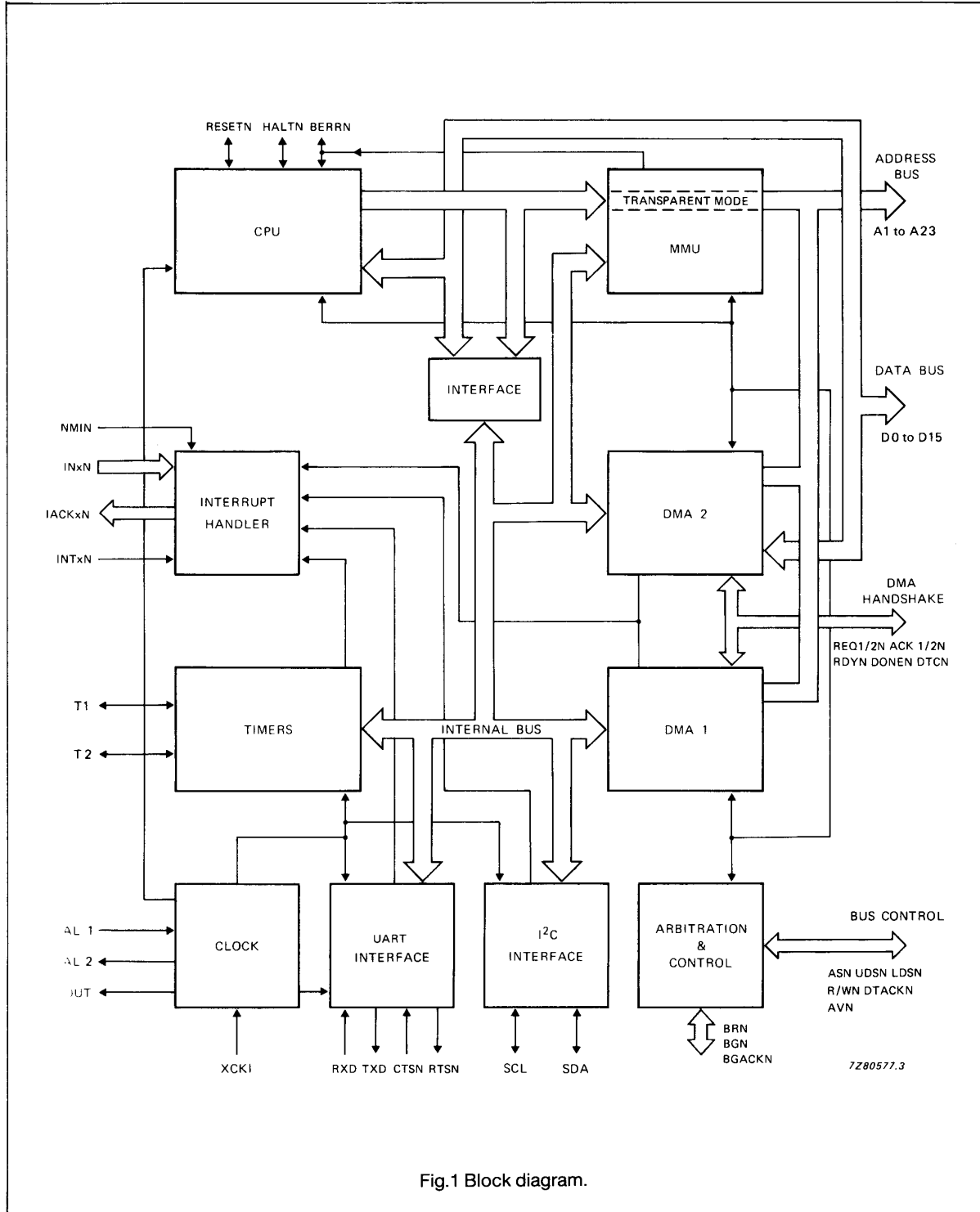


Fig.1 Block diagram.

16/32-bit microprocessor

SCC68070

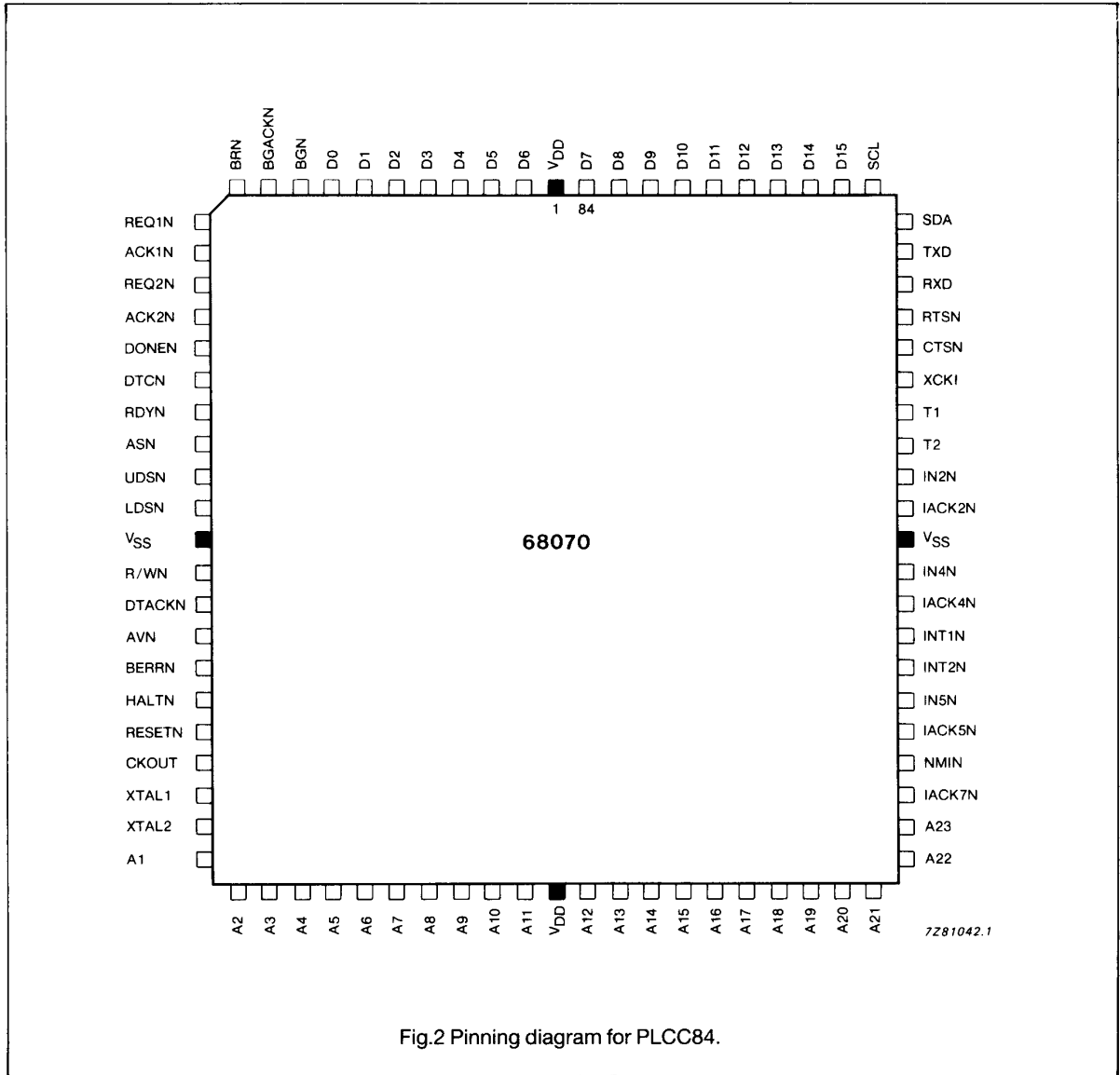
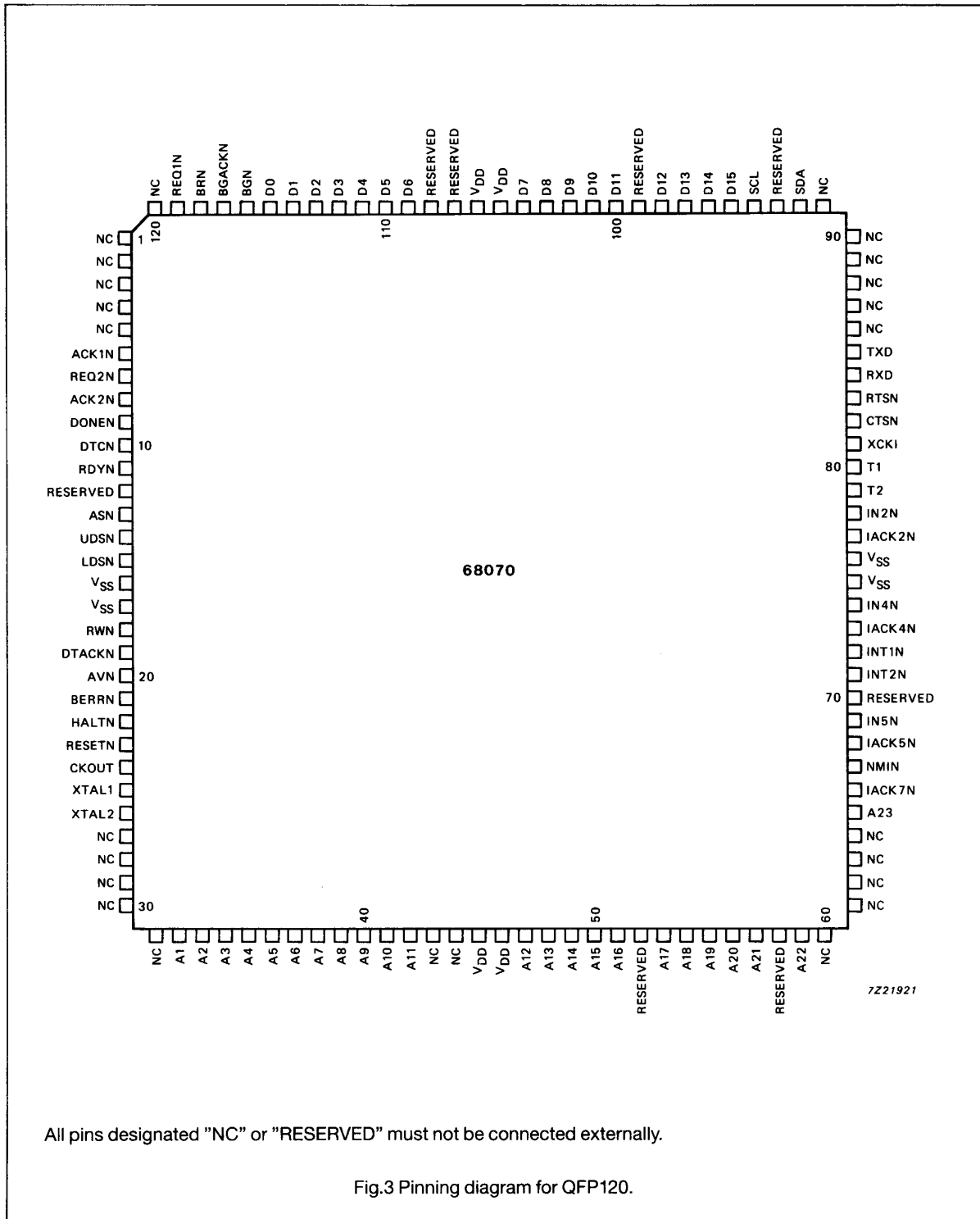


Fig.2 Pinning diagram for PLCC84.

16/32-bit microprocessor

SCC68070



16/32-bit microprocessor**SCC68070****Signal description (PLCC84)**

| MNEMONIC | TYPE | PIN NO. | FUNCTION |
|-----------------|-------------|----------------|--|
| A1 to A23 | O | 32-42, 44-55 | Address bus (active HIGH, 3-state). For direct addressing of 16 Mbytes of memory. |
| D0 to D15 | I/O | 8-2, 84-76 | Data bus (active HIGH, 3-state, bidirectional). 16-bit wide. |
| ASN | O | 19 | Address Strobe (active LOW, 3-state). Indicates a valid address on the bus. |
| LDSN | O | 21 | Lower Data Strobe (active LOW, 3-state). Indicates that: - For a WRITE cycle, the data is valid on the lower half of the data bus (D0 to D7). - For a READ cycle, the data is to be placed on the lower half of the bus (D0 to D7). |
| UDSN | O | 20 | Upper Data Strobe (active Low, 3-state). Indicates that: - For a WRITE cycle, the data is valid on the lower half of the data bus (D8 to D15). - For a READ cycle, the data is to be placed on the upper half of the bus (D8 to D15). |
| R/WN | O | 23 | Read (active HIGH)/ Write (active LOW). This controls the direction of data flow. |
| DTACKN | I | 24 | Data Transfer Acknowledge (active LOW). Asserted by the peripheral during CPU or DMA bus cycles when data is either received from or placed on the bus. If not asserted punctually, it causes the CPU or DMA controller to insert wait states. |
| BRN | I | 11 | Bus Request (active LOW). Asserted by wired-ORed external DMA devices that request bus ownership. |
| BGN | O | 9 | Bus Grant (active LOW). A daisy chain output that is asserted by the SCC68070 when the bus is granted by the CPU and the DMA does not have a bus request pending. |
| BGACKN | I/O | 10 | Bus Grant Acknowledge (active LOW, open drain). Asserted by any DMA device (internal or external) that has control of the bus. As long as this line is held LOW externally, the SCC68070 will hold the bus signals in the high impedance state. When BGACKN is released, the SCC68070 will have access to the bus. Interrupts cannot be serviced while BGACKN is held LOW. |
| RESETN | I/O | 28 | Reset (active LOW, open drain, bidirectional). If asserted externally together with the HALTN line, it will cause the processor to enter the Reset state. It is driven LOW by the processor when the Reset instruction resets external hardware. |
| HALTN | I/O | 27 | Halt (active LOW, open drain, bidirectional). If asserted externally together with RESETN, it causes the SCC68070 to enter the Reset state. If asserted alone, it will cause the CPU or DMA controller to stop after completion of the current bus cycle. If HALTN and BERRN are asserted together, the CPU will complete the current bus cycle, stop operation, and place all 3-state lines in their high impedance state until HALTN and BERRN have been released, and then it will re-run the same bus cycle. BERRN should be released before HALTN. As long as HALTN is held LOW all control signals are inactive and all 3-state lines are placed in their high impedance state. When the processor has stopped executing instructions (e.g. after a double bus fault) the processor drives this line LOW. |
| BERRN | I/O | 26 | Bus Error (active LOW, open drain). If this line is asserted during a bus cycle, it indicates that there was a fault in the bus cycle access. If asserted together with HALTN, the same bus cycle will re-run after both HALTN and BERRN have been released. If BERRN is asserted alone, the SCC68070 will start bus-error exception processing. BERRN is driven LOW by the SCC68070 when the MMU indicates a bus error. |

16/32-bit microprocessor**SCC68070**

| MNEMONIC | TYPE | PIN NO. | FUNCTION |
|---|-------------|-------------------|--|
| INT1N, INT2N | I | 61, 60 | Latched Interrupt inputs (active LOW). A LOW level of ≥ 1 clock pulse will be stored as a pending interrupt request. Priority levels are programmable. |
| IN2N, IN4N, IN5N | I | 66, 63, 59 | Decoded Interrupt priority inputs (active LOW). IN2N has the lower and IN5N has the higher priority. |
| NMIN | I | 57 | Non-maskable interrupt (level 7) (active LOW). While the other interrupts may be masked (disabled), this interrupt is always enabled. |
| IACK2N, IACK4N, IACK5N, IACK7N | O | 65, 62, 58, 56 | Decoded Interrupt acknowledge (active Low). Asserted during an interrupt acknowledge sequence to indicate to a peripheral that its interrupt request is being serviced. |
| AVN | I | 25 | Autovectored interrupts (active LOW). If held LOW during the interrupt acknowledge sequence, the processor calculates the appropriate vector from a fixed vector table. If kept HIGH, the peripheral must provide an 8-bit vector number. |
| V _{DD} | - | 1, 43 | Supply voltage + 5.0 V nominal. |
| V _{SS} | - | 22, 64 | Ground. |
| XTAL1, XTAL2 | I | 30, 31 | External crystal inputs . XTAL1 can be used as a clock input if an external clock generator is used. The crystal or external clock frequency is divided by 2 to obtain the internal clock and CKOUT signals. |
| CKOUT | O | 29 | Clock out . This is the reference from the internal system clock. |
| REQ1N, REQ2N | I | 12, 14 | DMA Request (active LOW). These are inputs from I/O devices requesting service from the DMA controller and causes it to request control of the bus. In burst mode, the inputs are level sensitive and the DMA controller releases the bus after REQ1N (or REQ2N) becomes active and the current DMA cycle is completed. In cycle-stealing mode, REQ1N or REQ2N inputs are triggered by a negative pulse. This pulse must occur at least one clock cycle before DTCN is asserted to ensure continuous transfer. |
| ACK1N, ACK2N | O | 13, 15 | DMA Request Acknowledge (active LOW). ACK1N (or ACK2N) is asserted by the DMA controller to indicate that it has acquired the bus and the requested device bus cycle is now beginning. It is active at the beginning of every device cycle together with ASN, and is deactivated at the end of every device bus cycle. |
| RDYN | I | 18 | Device Ready (active LOW). The requesting device asserts RDYN to indicate to the DMA controller that valid data has either been stored or put on the bus. If RDYN remains inactive, it indicates that the data has neither been stored nor put on the bus, causing the DMA controller to insert wait states. RDYN can be held LOW permanently if the device is fast enough, indicating that the device is always ready and so no wait states are required. RDYN is not monitored by Channel 2 in the dual address mode. |
| DTCN | O | 17 | Device Transfer Complete (active LOW, open drain). In DMA mode DTCN is asserted by the DMA controller to indicate to the device that the requested data transfer is complete. On a write-to-memory operation, it indicates that the data provided by the device has been stored successfully. On a read-from-memory operation, it indicates that the data from memory is present on the data bus and should be latched. |
| DONEN | I/O | 16 | Done (active LOW, open drain). With DONEN as an output, the DMA controller asserts it simultaneously with the ACK1N (or ACK2N) output to indicate to the device that the transfer count is zero and therefore, the DMA controller's operation is complete. If, as an input, DONEN is asserted by the device before the transfer count reaches zero, it causes the DMA controller to abort the operation and generate an interrupt request (if the interrupts are enabled). |

16/32-bit microprocessor**SCC68070**

| MNEMONIC | TYPE | PIN NO. | FUNCTION |
|-----------------|-------------|----------------|---|
| SCL | I/O | 75 | Serial Clock (open drain). SCL is the clock signal for the I ² C-bus operation. It is either driven by the SCC68070 when the I ² C interface is in the master mode, or is the clock input if the I ² C interface is in the slave mode. |
| SDA | I/O | 74 | Serial Data (open drain). SDA is the data signal for the I ² C-bus. |
| T1, T2 | I/O | 68, 67 | Timers 1 and 2 (3-state). These are the I/O signals for the capture timers of channels 1 and 2 respectively. They can be programmed as either outputs for pulses or inputs for count cycles and events. |
| RXD | I | 72 | Receive Data . RXD is the data input for the UART serial interface. |
| TXD | O | 73 | Transmit Data . TXD is data output for the UART serial interface. |
| RTSN | O | 71 | Request To Send (active LOW). This output of the UART serial interface indicates that the receiver is ready to accept data on the RXD line. |
| CTSN | I | 70 | Clear To Send (active LOW). This input to the UART serial interface indicates that the remote receiving device is ready. RTSN and CTSN can be connected together if no control lines are needed. |
| XCKI | I | 69 | External clock . When selected, XCKI is the clock input for the UART serial interface. This signal can be used either: - to generate special baud rates or, - when a crystal frequency other than 19.6608 MHz is used by the SCC68070, an external clock of 4.9152 MHz (or 9.8304 for 38200 bauds) can be connected to this input to generate the standard baud rates. |

Note

The signal descriptions given for the PLCC84 package also apply to the QFP120 package. However, the pinning arrangement for the QFP120 is different, as can be seen in Fig.3.

16/32-bit microprocessor

SCC68070

CPU FUNCTIONAL DESCRIPTION

General

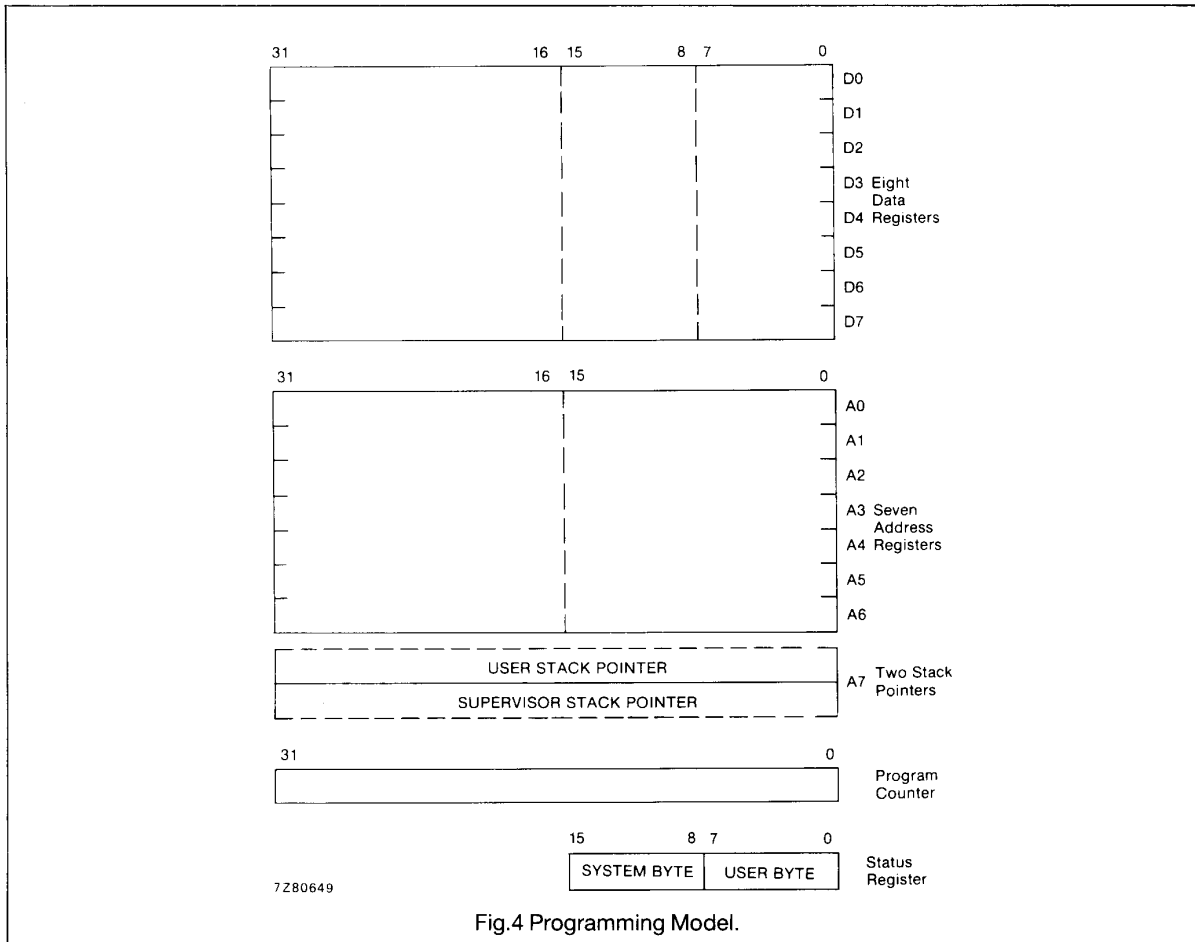
The CPU of the SCC68070 is software compatible with the 68000, consequently programs written for the 68000 will run on the SCC68070 unchanged. However, for certain applications the following differences between the processors should be noted:

- Differences exist in the exception error processing since the SCC68070 can provide full bus-error recovery.
- The timing is different because of the SCC68070's new architecture and technology. Although the bus timing is similar to the 68000, instruction execution timing is completely different. For execution timing see Tables 7 to 19.

Programming model and data organization

The programming model is identical to that of the 68000 and is shown in Fig.4. It contains seventeen 32-bit registers, a 32-bit Program Counter and a 16-bit Status Register (see Fig.6). The first eight registers (D0 to D7) are used for data registers for byte, word and long-word operations. The second group of registers (A0 to A6) and the system stack pointer (A7) can be used as software stack pointers and base address registers. In addition, these registers can be used for word and long-word address operations. All seventeen registers can be used as index registers.

The SCC68070 supports bit data, integer data of 8, 16 and 32 bits, 32-bit addresses and BCD data. Each data type is arranged in the memory as shown in Fig.5.



16/32-bit microprocessor

SCC68070

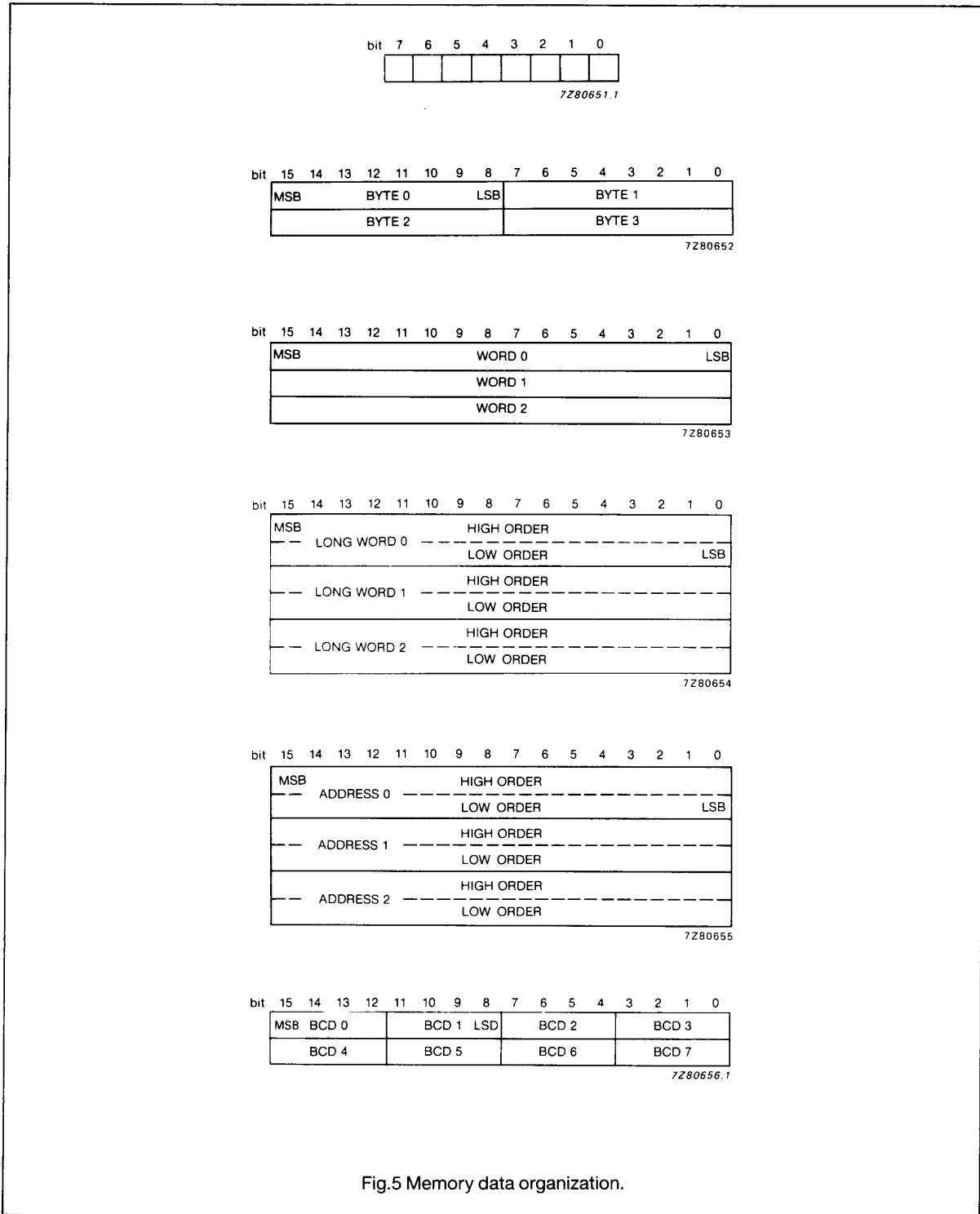
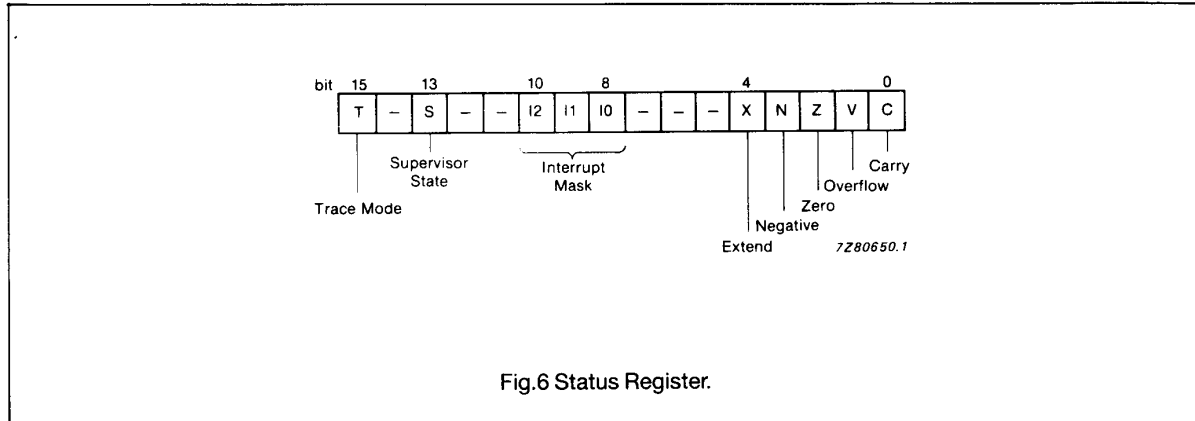


Fig.5 Memory data organization.

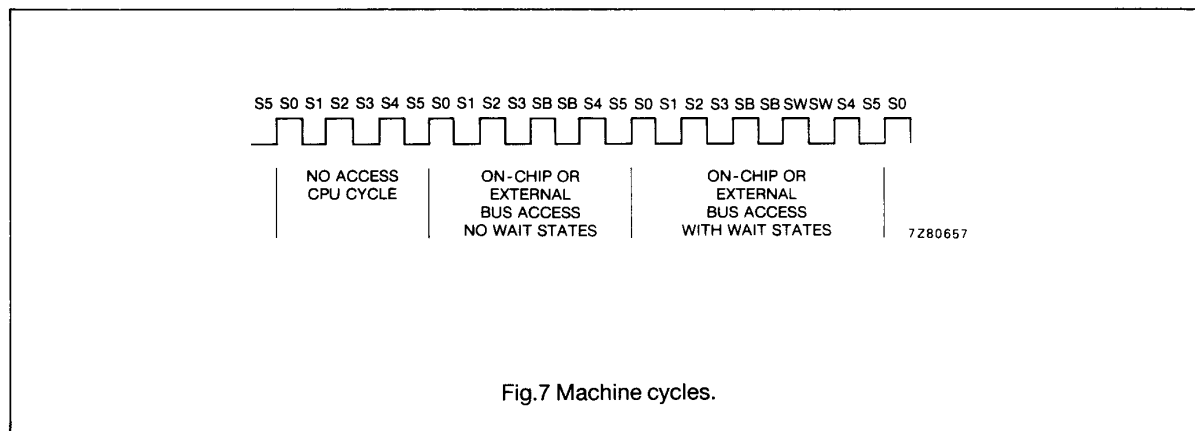
16/32-bit microprocessor

SCC68070



Internal and external operation

The SCC68070 operates with a maximum internal clock frequency of 17.5 MHz and a minimum of 4 MHz. Each clock cycle is divided into 2 states. A non-access machine cycle has 3 clock cycles or 6 states (S0 to S5). An on-chip or external bus access normally consists of 3 clock cycles plus 1 clock cycle (2 SB states). When DTACKN is not asserted, indicating that data has not been received or put on the bus, wait states (SW) are inserted in multiples of 2.



16/32-bit microprocessor

SCC68070

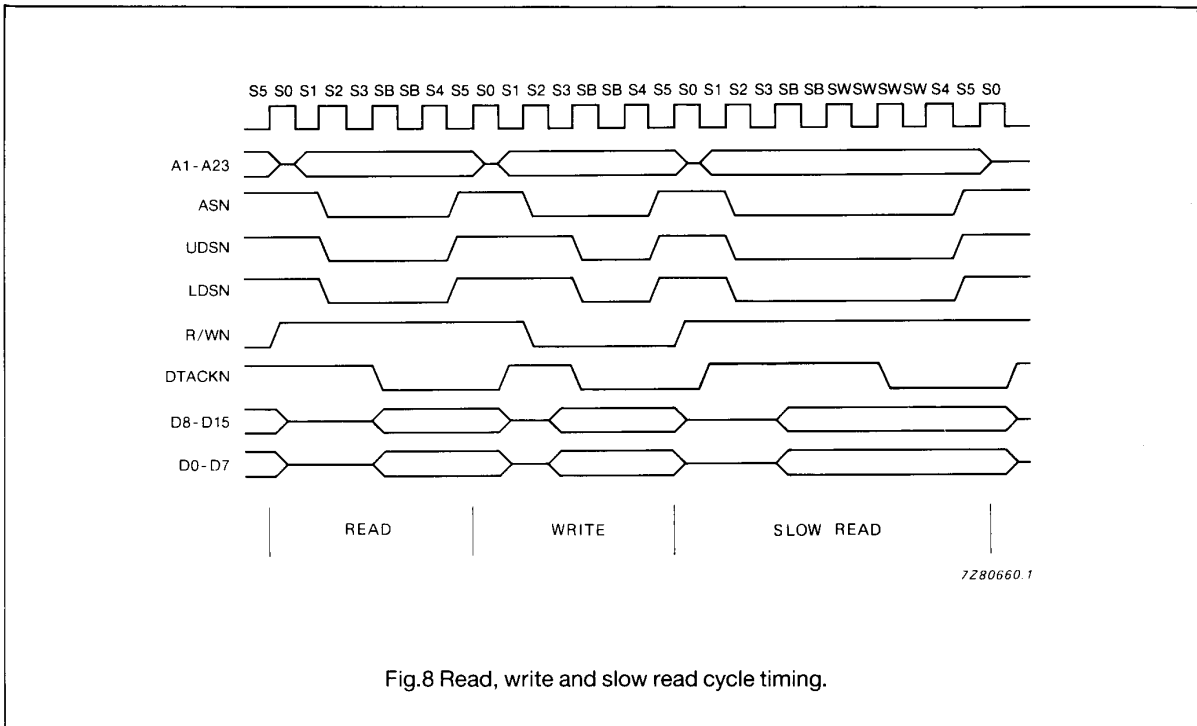


Fig.8 Read, write and slow read cycle timing.

Bus timing

Bus cycles in the SCC68070 are similar to those of a 68000 running at a CKOUT frequency. However, if the DTACKN signal is not asserted by the time the SCC68070 is ready to transmit or receive data, it will insert wait cycles. Upper and lower data strobes (UDSN and LDSN) are asserted independently with respect to the type of transfer (low byte - LDSN asserted, high byte - UDSN asserted, and word - both strobes asserted).

Bus arbitration

Because a DMA controller is integrated on the SCC68070 as a possible bus master, the bus arbitration needs a priority protocol. This is done by a daisy-chain using the BUS GRANT (BGN) of the CPU such that Channel 1 of the DMA controller has highest priority, followed by Channel 2 and then the external devices. The CPU grants bus acquisition and therefore has lowest priority. Once the DMA controller has submitted the internal bus grant to an outside master it will not interrupt the line until BUS GRANT ACKNOWLEDGE (BGACKN) has been negated externally. If the DMA controller has a DMA request

pending, it will acquire the bus as soon as the external device has negated BGACKN. In this event, it will not submit BGN to an outside master, even if the prospective masters BRN signal had been asserted before the DMA controller's pending request.

Processing states

The CPU is always in one of three processing states: normal, exception or halted.

The normal processing state is that associated with instruction execution; the memory references are to fetch instructions and operands, and to store results. A special case of the normal state is the stopped state which the processor enters when a STOP instruction is executed. In this state, no further memory accesses are made by the CPU.

The exception processing state is associated with interrupts, trap instructions, tracing and other exceptional conditions. The exception may be generated internally by an instruction or an unusual condition arising during the execution of an instruction. Externally, exception processing can be forced by an interrupt, by bus error or by a reset. Exception processing is designed to provide

16/32-bit microprocessor

SCC68070

an efficient context switch so that the processor can handle unusual conditions.

The halted processing state is an indication of a catastrophic hardware failure. For example, if during exception processing of a bus error another bus error occurs, the CPU assumes that the system is unusable and halts. Only an external reset can restart a halted processor. Note that a CPU in the stopped state is not in the halted state or vice versa.

The processor can work in the 'user' or 'supervisor' state determined by the state of the S-bit in the Status Register. Accesses to the on-chip peripherals must be in the supervisor state.

Exception processing

Exception processing occurs in four steps. First, a copy is made of the contents of the Status Register and then the S-bit is asserted, putting the processor into the privileged supervisor state. Second, the vector number of the exception is determined; this is used to generate the address of the exception routine. The next step saves the current processor status. Copies of the current program counter, the Status Register and the Format plus vector

number are saved on the supervisor stack using the supervisor stack pointer. Finally the contents of the exception vector location are fetched and loaded into the Program Counter and the exception handling routine starts normal instruction execution.

Exception vectors

Exception vectors are memory locations from which the CPU fetches the address of a routine that will handle that exception. All exception vectors are 2 words in length (see Fig.9) except the reset vector, which is made up of 4 words. All exception vectors are contained in the supervisor data space. When the reset vector is fetched after a RESETN, the MMU is disabled and the reset vector is located at physical address 0. A vector number is an 8-bit number that, when multiplied by 4, gives the address of an exception vector. Vector numbers are generated internally or externally depending on the cause of the exception. During the interrupt acknowledge bus cycle, an external peripheral may send the CPU an 8-bit vector number (see Fig.10) on the data bus lines D0 to D7. The CPU translates the vector number into the full 24-bit address as shown in Fig.11. The memory layout for the exception vectors is given in Table 1.

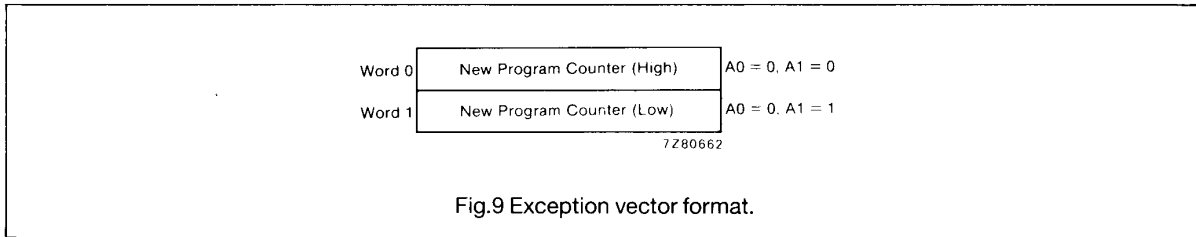


Fig.9 Exception vector format.

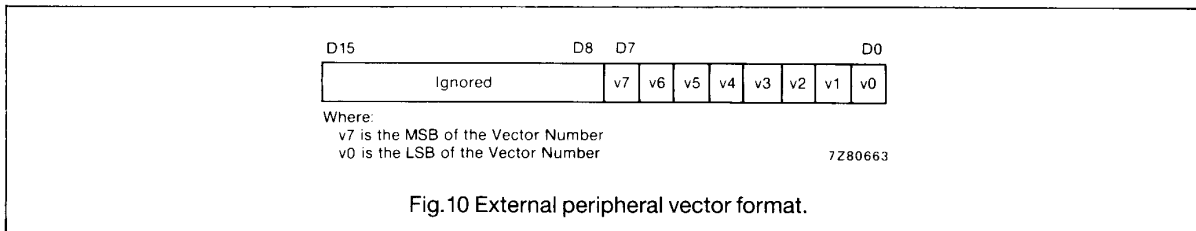


Fig.10 External peripheral vector format.

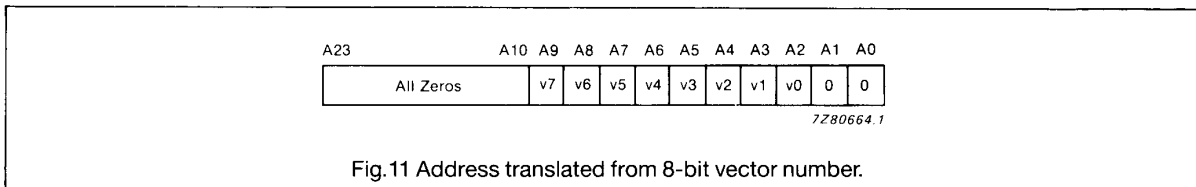


Fig.11 Address translated from 8-bit vector number.

16/32-bit microprocessor**SCC68070****Table 1** Exception vector assignment.

| VECTOR NOS. | DEC | HEX | ASSIGNMENT |
|-------------|-----|-----|--------------------------------------|
| 0 | 0 | 000 | Reset:initial SSP |
| 1 | 4 | 004 | Reset:initial PC |
| 2 | 8 | 008 | Bus error |
| 3 | 12 | 00C | Address error |
| 4 | 16 | 010 | Illegal instruction |
| 5 | 20 | 014 | Zero divide |
| 6 | 24 | 018 | CHK instruction |
| 7 | 28 | 01C | TRAPV instruction |
| 8 | 32 | 020 | Privilege violation |
| 9 | 36 | 024 | Trace |
| 10 | 40 | 028 | Line 1010 emulator |
| 11 | 44 | 02C | Line 1111 emulator |
| 12* | 48 | 030 | (Unassigned, reserved) |
| 13* | 52 | 034 | (Unassigned, reserved) |
| 14 | 56 | 038 | Format error |
| 15 | 60 | 03C | Uninitialized vector interrupt |
| 16-23* | 64 | 040 | (Unassigned, reserved) |
| - | 95 | 05F | - |
| 24 | 96 | 060 | Spurious interrupt |
| 25 | 100 | 064 | Level 1 interrupt autovector |
| 26 | 104 | 068 | Level 2 interrupt autovector |
| 27 | 108 | 06C | Level 3 interrupt autovector |
| 28 | 112 | 070 | Level 4 interrupt autovector |
| 29 | 116 | 074 | Level 5 interrupt autovector |
| 30 | 120 | 078 | Level 6 interrupt autovector |
| 31 | 124 | 07C | Level 7 interrupt autovector |
| 32-47 | 128 | 080 | TRAP instruction vectors |
| - | 191 | 0BF | - |
| 48-56* | 192 | 0C0 | (Unassigned, reserved) |
| - | 227 | 0E3 | - |
| 57 | 228 | 0E4 | Level 1 on-chip interrupt autovector |
| 58 | 232 | 0E8 | Level 2 on-chip interrupt autovector |
| 59 | 236 | 0EC | Level 3 on-chip interrupt autovector |
| 60 | 240 | 0F0 | Level 4 on-chip interrupt autovector |
| 61 | 244 | 0F4 | Level 5 on-chip interrupt autovector |
| 62 | 248 | 0F8 | Level 6 on-chip interrupt autovector |
| 63 | 252 | 0FC | Level 7 on-chip interrupt autovector |
| 64-255 | 256 | 100 | User interrupt vectors |

* Vectors 12, 13, 16 to 23, and 48 to 56 are reserved for future enhancements. No user peripheral devices should be assigned to these numbers.

16/32-bit microprocessor

SCC68070

Multiple exceptions

As two or more exceptions can occur simultaneously, exceptions are grouped in order of priority; as is shown in Table 2.

Table 2 Exception grouping and priority.

| GROUP | EXCEPTION | PROCESSING |
|-------|--|--|
| 0 | RESET ADDR.ERROR BUS ERROR | Exception processing begins at the next machine cycle |
| 1 | TRACE INTERRUPT ILLEGAL PRIVILEGE | Exception processing begins before the next instruction |
| 2 | TRAP, TRAPV, CHK, ZERO, DIVIDE, FORMAT ERROR | Exception processing is started through normal instruction execution |

Reset operation

When the CPU executes a RESET instruction, the RESETN signal is driven LOW for 146 clock cycles to reset internal and external peripherals. The CPU itself is not affected but all on-chip peripherals are reset. When both the RESETN and HALTN signals are driven LOW by an external device, the CPU and on-chip peripherals are reset. The CPU responds by reading the reset vector table entry (vector number zero, address 000000H) and loads it to the Supervisor Stack Pointer (SSP). Vector table entry number one (at address 000004H) is read next and loaded into the Program Counter (PC). The CPU then initializes the Status Register (SR) to an interrupt level of seven and instruction execution is started (see Fig.12). All 3-state output signals are placed in the high-impedance state for as long as the RESETN and HALTN signals are externally driven. When the RESETN and HALTN signals are released, the CPU will execute 4 read cycles after start-up time, to load the SSP High, SSP Low, PC High and PC Low. Then the first instruction is fetched and executed. The HALTN signal must be driven LOW at the same time as the RESETN signal. The SCC68070 will only start to read the stack and the initial Program Counter after both signals have been released. RESETN should not be released after HALTN. When V_{DD} is initially applied to the SCC68070, an external RESET must be applied to the RESETN pins for at least 100 ms.

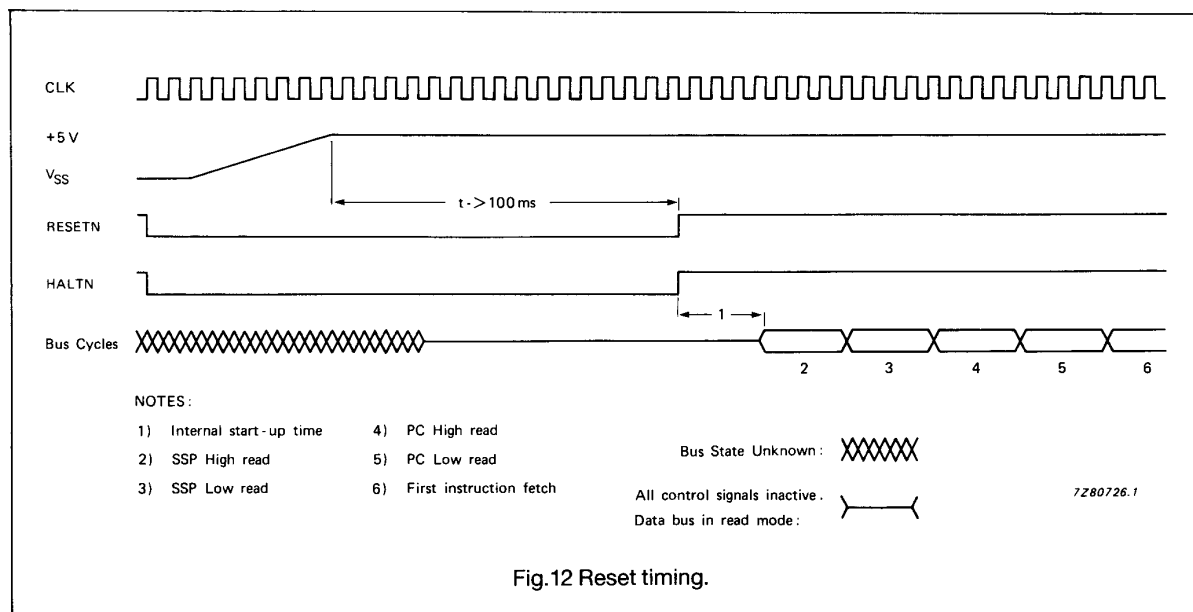


Fig.12 Reset timing.

16/32-bit microprocessor

SCC68070

Bus error processing

Like the 68000 the SCC68070 uses the BUS ERROR (BERRN) and the HALT signals to distinguish between two bus-error handling routines. If the BERRN and HALTN signals are asserted, the SCC68070 will re-run the last bus cycle as soon as both BERRN and HALTN lines are released. This is valid for both the CPU and DMA controller. BERRN should become inactive before HALTN (see Fig.13). If just the BERRN signal is asserted, bus error exception processing is entered. If the DMA controller is the affected master, it will:

- stop the DMA service after the current bus cycle is terminated
- release the bus
- set the BERRN bit in the status word, and
- send an interrupt if the INTERRUPT ENABLE bit in CCR (3) was set.

The address counter reflects the address of the faulty bus cycle, while the transfer counter indicates the number of successful transfers.

If the CPU is the bus master, it will enter the bus error exception processing after the current bus cycle is terminated and BERRN has been released. Since the architecture of the SCC68070 differs from the 68000, it handles bus errors in a different manner. Unlike the 68000, the SCC68070 enables full recovery from bus errors.

The procedure follows the usual sequence of steps:

- the status register is copied internally
- the supervisor state is entered
- the trace state is turned off, and
- the vector number is generated to refer to the bus error vector.

To save more of the context, additional information is stored on the stack as follows:

- The program counter and a copy of the status register are saved.
- A format word containing a special bit configuration for the SCC68070 long stack format, and the vector number of the exception, in this case, the bus error vector, is stacked.
- Besides other internal information, the processor saves the address which was being accessed by the aborted bus cycle.
- Specific information about the access is either saved or is retrievable from stacked information.
- A special status word is saved to determine the state of the (internal) function codes, the source of the bus error (MMU or External), and whether the error occurred during a read or write cycle. A RERUN bit in this special status word has to be set to suppress a retry of the faulty bus cycle on Return from Exception (RTE).
- The instruction registers and temporary registers are saved.

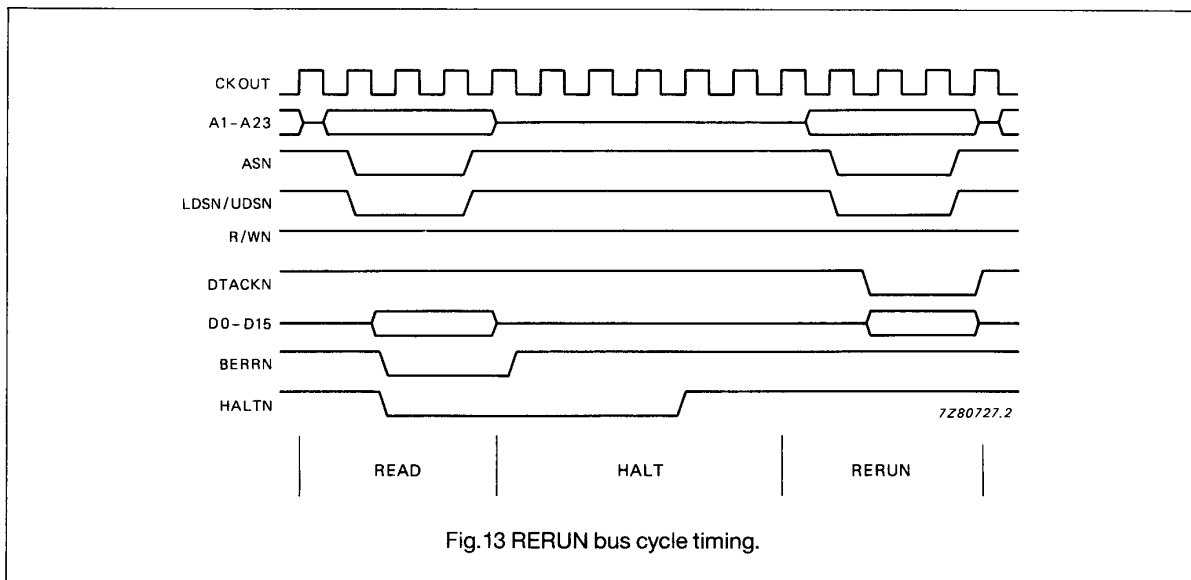


Fig.13 RERUN bus cycle timing.

16/32-bit microprocessor

SCC68070

Stack format

The stacking operation for the exception processing is similar to the 68010 (rather than the 68000) however, the information stored is not the same due to the different architecture. To handle this, the following changes from the 68000 have been made:

- The stack format has changed.
- The minimum number of words put in or restored from the stack is 4 (short stack format) which is 68010 compatible; not 3 as for the 68000.
- The RTE instruction decides (with the aid of the 4 format bits) whether more information has to be

restored. The SCC68070 long format is used for bus error and address error exceptions, all other exceptions use the short format.

- If another format code other than one of the two listed above is detected during the restore action, a FORMAT ERROR occurs.

If the user wants to finish the instruction in which the bus or address error occurred (modification of the stack), the SCC68070 format must be used on RTE. If no changes are required to the stack during exception processing, the stack format is transparent to the user.

The stack format is shown in Fig.14 and the special status word shown in Fig.15.

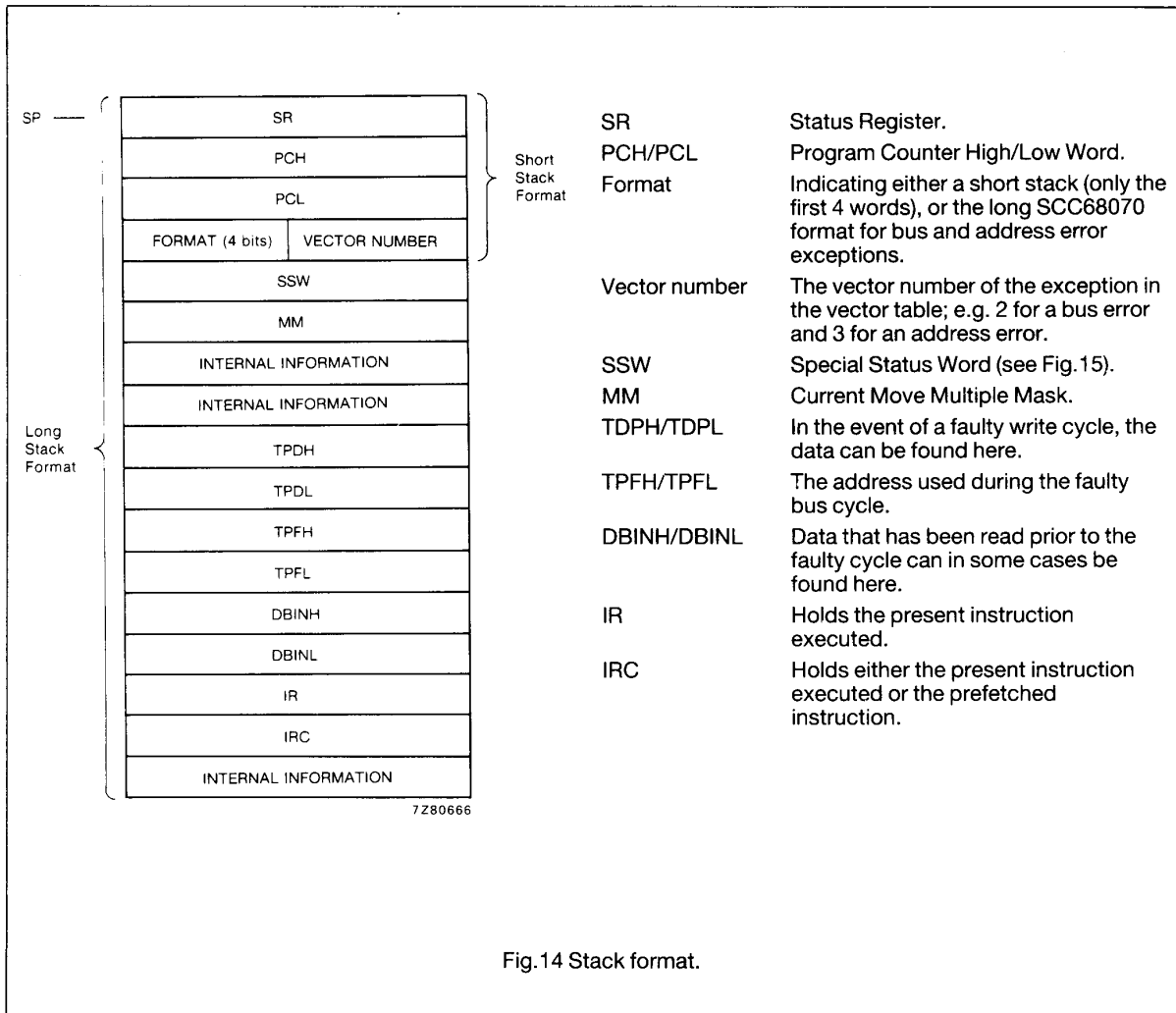
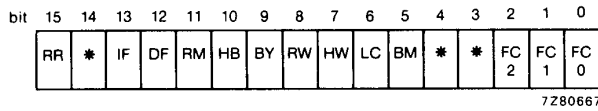


Fig.14 Stack format.

16/32-bit microprocessor

SCC68070



- RR Rerun. By default this bit is set to '0'. If set to '1' the CPU will not re-run the faulty bus cycle on return from exception (RTE).
- * Undefined (reserved).
- IF The faulty cycle was an instruction fetch.
- DF The faulty cycle was a data fetch.
- RM The error occurred during a read-modify-write cycle.
- HB High Byte.
- BY The faulty cycle was a byte transfer.
- RW Read/write cycle.
- HW High Word.
- LC The faulty cycle was during a long-word access.
- BM The bus error was caused by the on-chip MMU.
- FC2,1,0 These bits hold the internal function code during the faulty bus cycle. The function codes are same as for the 68000 and affect the status of the CPU during the faulty bus cycle as follows.

| FC2 | FC1 | FC0 | |
|-----|-----|-----|-----------------------|
| 0 | 0 | 0 | reserved |
| 0 | 0 | 1 | user data |
| 0 | 1 | 0 | user program |
| 0 | 1 | 1 | reserved |
| 1 | 0 | 0 | reserved |
| 1 | 0 | 1 | supervisor data |
| 1 | 1 | 0 | supervisor program |
| 1 | 1 | 1 | interrupt acknowledge |

Fig.15 Special status word.

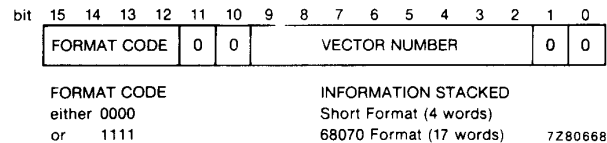
16/32-bit microprocessor**SCC68070**

Fig.16 Vector number and format code.

Interrupt processing

The SCC68070 interrupt handling follows the same basic rules as the 68000. However, the following changes have been made to simplify system development:

- the IPL signals have been replaced by decoded interrupt signals IN2N, IN4N, IN5N and NMIN representing levels 2, 4, 5 and 7 respectively
- each of the interrupts has a separate acknowledge signal IACK2N, IACK4N, IACK5N and IACK7N
- two latched interrupt inputs (INT1N and INT2N) have programmable priority levels. They have no interrupt acknowledge signal and are always served by autovectoring with vectors located in the "on-chip" entry in the vector table
- interrupt priority levels IPL1, IPL3 and IPL6 are not available externally, unless programmed into INT1N or INT2N
- if autovectoring is desired for IN2N, IN4N, IN5N or NMIN, the AUTOVECTOR request signal (AVN) must be asserted during the interrupt acknowledge routine.
- if more than one interrupt line is asserted at the same time, the one with the highest priority will be serviced first
- to ensure being recognized, an interrupt signal IN2N, IN4N, IN5N and NMIN must stay asserted until acknowledged by its IACK1N or IACK2N signal
- interrupts with a priority level equal to or less than the priority level actually running will not be accepted
- during the acknowledge cycle of an interrupt, the IPL bits of the status register are set to the priority level of the acknowledged interrupt.

If the priority of the interrupt pending is greater than the current processor priority then:

- the exception processing sequence is started
- a copy of the status register is saved
- the privilege level is set to supervisor state
- tracing is suppressed
- the priority level of the processor is set to that of the interrupt being acknowledged.

The processor then gets the vector number from the interrupting device, classifies it as an interrupt acknowledge, and displays the interrupt level number being acknowledged on the address bus. During the interrupt acknowledge cycle ASN is asserted to indicate that the bus is occupied but LDSN is not asserted (note: the 68000 asserts both ASN and LDSN). This is done to simplify the address decoding circuitry of the memory. Acknowledge cycle decoding (by the interrupting device) is done using the IACK1N (or IACK2N) signal instead.

If autovectoring is requested by the internal logic, the processor generates a vector number internally that corresponds to the interrupt level number. Then, if a bus error is indicated by the external logic, the interrupt is treated as spurious and the vector number that was generated will refer to the spurious interrupt vector.

Priority level 7 is a special case and its interrupts cannot be inhibited by the interrupt priority mask thus providing a Non-Maskable Interrupt (NMIN) capability. An interrupt is generated each time the interrupt request level changes from a lower level to level 7.

16/32-bit microprocessor

SCC68070

If external and on-chip peripherals are programmed to the same interrupt priority level, an on-chip daisy-chain defines the priority as follows:

| EXTERNAL INTERRUPTS | PRIORITY LEVEL |
|---------------------|------------------|
| IN1N, IN2N | highest priority |
| INT1N | |
| INT2N | |
| TIMER | |
| UART RX | |
| UART TX | |
| I ² C | |
| DMA CH1 | |
| DMA CH2 | |
| | lowest priority |

The two latched interrupt inputs INT1N and INT2N have a common Latched Interrupt priority level register (LIR) as shown in Fig.17.

bit 7 6 5 4 3 2 1 0

| | | | | | | | |
|-------|-----|-----|-----|-------|-----|-----|-----|
| INT1N | | | | INT2N | | | |
| PIR | IPL | IPL | IPL | PIR | IPL | IPL | IPL |
| IN1 | 2 | 1 | 0 | IN2 | 2 | 1 | 0 |

7280671.3

INT1N and INT2N refer to the external pins.

IPL Interrupt priority level of interrupts connected externally to pins INT1N or INT2N. IPL2 is the MSB and IPL0 the LSB. All values are positive true; IPL = 111 represents priority level 7 and IPL = 000 will inhibit the interrupts.

PIR Pending Interrupt Reset, when set to '1' any pending interrupts of the respective interrupt input will be reset and the inputs must be toggled again to create another interrupt. Note that this does not reset the interrupting status of a connected peripheral and further, if the IPL bits are changed without setting the PIR bit then spurious interrupts may occur. Reading the PIR bits will return zeros.

note: Initially and after RESET, all LIR bits are cleared to zero.

Fig.17 Latched Interrupt priority level Register (LIR).

On-chip addresses

All memory locations on the peripheral side of the on-chip interface can only be accessed in SUPERVISOR mode; i.e. the S-bit in the Processor Status Word (PSW) is set to '1'. If in USER mode (S = '0'), an external bus access is performed. All registers on the peripheral side of the interface are memory mapped separately from the 68070's external 16 Mbyte memory map. The on-chip address space is decoded by the two MSBs of the 32-bit internal address and the S-bit of the processor status word, as shown in Table 3. The address map of all the on-chip peripherals is given in Table 4.

Table 3 A31, A30 and S-bit decoding.

| S | A31 | A30 | |
|---|-----|-----|----------|
| X | 0 | 0 | external |
| X | 0 | 1 | external |
| 1 | 1 | 0 | internal |
| 0 | 1 | 0 | external |
| X | 1 | 1 | external |

Table 4 Address map of on-chip peripherals.

| ADDRESS RANGE (HEX) | DESCRIPTION |
|----------------------|----------------------------|
| 00000000 to 7FFFFFFF | off-chip |
| 80000000 to 80001000 | on-chip, reserved |
| 80001001 | LIR priority level |
| 80001002 to 80002000 | on-chip, reserved |
| 80002001 to 80002009 | I ² C interface |
| 8000200A to 80002010 | on-chip, reserved |
| 80002011 to 8000201B | UART interface |
| 8000201C to 8000201F | on-chip, reserved |
| 80002020 to 80002029 | TIMER |
| 8000202A to 80002024 | on-chip reserved |
| 80002045 | PICR1 |
| 80002046 | on-chip, reserved |
| 80002047 | PICR2 |
| 80002048 to 80003FFF | on-chip, reserved |
| 80004000 to 8000406D | DMA controller |
| 8000406E to 80007FFF | on-chip, reserved |
| 80008000 to 8000807F | MMU |
| 80008080 to BFFFFFFF | on-chip, reserved |
| C0000000 to FFFFFFFF | off-chip |

16/32-bit microprocessor

SCC68070

Clock circuitry

The clock signals required by the SCC68070 are generated from one master oscillator. Dividing this frequency by two gives the clock signals for the CPU, MMU and DMA. Dividing the master oscillator frequency by four provides the clock frequency for the UART and I²C interface. Dividing the master oscillator frequency by 192 gives the clock for the Timer.

When the master oscillator frequency differs from 19.6608 MHz and standard baud rates are required from the UART, it is possible to clock the UART separately by supplying externally a 4.9152 or 9.8304 MHz signal on the XCKI pin (see Fig.18). The Timer will still have a period of 192 times that of the master oscillator period. The speed of the I²C interface is programmable and may need to be programmed with a different division factor.

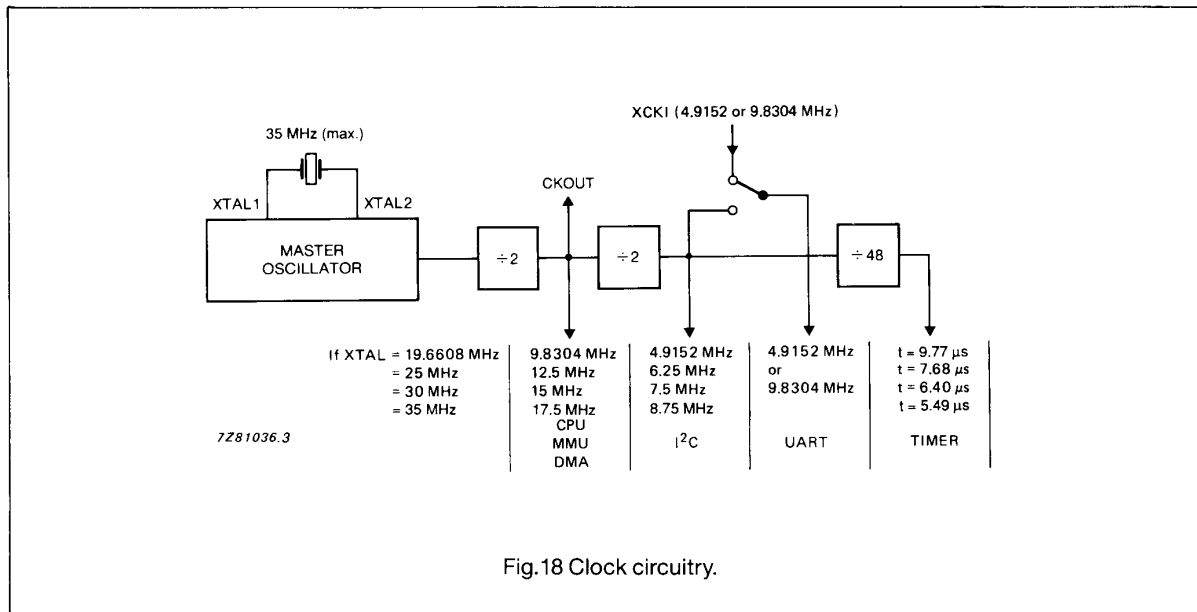


Fig.18 Clock circuitry.

16/32-bit microprocessor**SCC68070****INSTRUCTION SET AND ADDRESSING MODES**

The SCC68070 is completely code compatible with the 68000 consequently programs developed for the 68000 will run on the SCC68070. This applies to both source and

object code. The instruction set was designed to minimize the number of mnemonics that the programmer has to remember. Tables 5 and 6 give an overview of the instruction set and of the different addressing modes.

Table 5 Instruction set.

| MNEMONIC | DESCRIPTION | OPERATION | CONDITION CODES | | | | |
|----------|--------------------------------------|--|-----------------|---|---|---|---|
| | | | X | N | Z | V | C |
| ABCD | Add Decimal with Extend | $(\text{Destination})_{10} + (\text{Source})_{10} \rightarrow \text{Destination}$ | * | U | * | U | * |
| ADD | Add Binary | $(\text{Destination}) + (\text{Source}) \rightarrow \text{Destination}$ | * | * | * | * | * |
| ADDA | Add Address | $(\text{Destination}) + (\text{Source}) \rightarrow \text{Destination}$ | - | - | - | - | - |
| ADDI | Add Immediate | $(\text{Destination}) + \text{Immediate Data} \rightarrow \text{Destination}$ | * | * | * | * | * |
| ADDQ | Add Quick | $(\text{Destination}) + \text{Immediate Data} \rightarrow \text{Destination}$ | * | * | * | * | * |
| ADDX | Add Extended | $(\text{Destination}) + (\text{Source}) + X \rightarrow \text{Destination}$ | * | * | * | * | * |
| AND | AND Logical | $(\text{Destination}) \wedge (\text{Source}) \rightarrow \text{Destination}$ | - | * | * | 0 | 0 |
| ANDI | AND Immediate | $(\text{Destination}) \wedge \text{Immediate Data} \rightarrow \text{Destination}$ | - | * | * | 0 | 0 |
| ASL, ASR | Arithmetic Shift | (Destination) Shifted by $\langle \text{count} \rangle \rightarrow \text{Destination}$ | * | * | * | * | * |
| BCC | Branch Conditionally | If CC then $\text{PC} + d \rightarrow \text{PC}$ | - | - | - | - | - |
| BCHG | Test a Bit and Change | $\sim(\langle \text{bit number} \rangle \text{ OF Destination}) \rightarrow Z$ $\sim(\langle \text{bit number} \rangle \text{ OF Destination}) \rightarrow$ $\langle \text{bit number} \rangle \text{ OF Destination}$ | - | - | * | - | - |
| BCLR | Test a Bit and Clear | $\sim(\langle \text{bit number} \rangle \text{ OF Destination}) \rightarrow Z$ $0 \rightarrow \langle \text{bit number} \rangle \text{ OF Destination}$ | - | - | * | - | - |
| BRA | Branch Always | $\text{PC} + d \rightarrow \text{PC}$ | - | - | - | - | - |
| BSET | Test a Bit and Set | $\sim(\langle \text{bit number} \rangle \text{ OF Destination}) \rightarrow Z$ $1 \rightarrow \langle \text{bit number} \rangle \text{ OF Destination}$ | - | - | * | - | - |
| BSR | Branch to Subroutine | $\text{PC} \rightarrow \text{SP}@-; \text{PC} + d \rightarrow \text{PC}$ | - | - | - | - | - |
| BTST | Test a Bit | $\sim(\langle \text{bit number} \rangle \text{ OF Destination}) \rightarrow Z$ | - | - | - | - | - |
| CHK | Check Register against Bounds | If $\text{Dn} < 0$ or $\text{Dn} > \langle \text{ea} \rangle$ then TRAP | - | * | U | U | U |
| CLR | Clear an Operand | $0 \rightarrow \text{Destination}$ | - | 0 | 1 | 0 | 0 |
| CMP | Compare | $(\text{Destination}) - (\text{Source})$ | - | * | * | * | * |
| CMPA | Compare Address | $(\text{Destination}) - (\text{Source})$ | - | - | * | * | * |
| CMPI | Compare Immediate | $(\text{Destination}) - \text{Immediate Data}$ | - | * | * | * | * |
| CMPM | Compare Memory | $(\text{Destination}) - (\text{Source})$ | - | * | * | * | * |
| DBCC | Test Condition, Decrement and Branch | If CC then $\text{Dn} - 1 \rightarrow \text{Dn}$; if $\text{Dn} \neq -1$ then $\text{PC} + d \rightarrow \text{PC}$ | - | - | - | - | - |
| DIVS | Signed Divide | $(\text{Destination}) / (\text{Source}) \rightarrow \text{Destination}$ | - | * | * | * | 0 |
| EOR | Exclusive OR Logical | $(\text{Destination}) \oplus (\text{Source}) \rightarrow \text{Destination}$ | - | * | * | 0 | 0 |
| EORI | Exclusive OR Immediate | $(\text{Destination}) \oplus \text{Immediate Data} \rightarrow \text{Destination}$ | - | * | * | 0 | 0 |
| EXG | Exchange Register | $\text{Rx} \leftrightarrow \text{Ry}$ | - | - | - | - | - |
| EXT | Sign Extend | $(\text{Destination}) \text{Sign-extended} \rightarrow \text{Destination}$ | - | * | * | 0 | 0 |
| JMP | Jump | $\text{Destination} \rightarrow \text{PC}$ | - | - | - | - | - |
| JSR | Jump to Subroutine | $\text{PC} \rightarrow \text{SP}@-; \text{Destination} \rightarrow \text{PC}$ | - | - | - | - | - |
| LEA | Load Effective Address | $\text{Destination} \rightarrow \text{An}$ | - | - | - | - | - |
| LINK | Link and Allocate | $\text{AN} \rightarrow \text{SP}@-; \text{SP} \rightarrow \text{An}; \text{SP} + d \rightarrow \text{SP}$ | - | - | - | - | - |

16/32-bit microprocessor

SCC68070

| MNEMONIC | DESCRIPTION | OPERATION | CONDITION CODES | | | | |
|--------------|--------------------------------------|--|-----------------|---|---|---|---|
| | | | X | N | Z | V | C |
| LSL, LSR | Logical Shift | (Destination)Shifted by <count>→Destination | * | * | * | 0 | * |
| MOVE | Move Data from Source to Destination | (Source)→Destination | - | * | * | 0 | 0 |
| MOVE to CCR | Move to Condition Code | (Source)→CCR | * | * | * | * | * |
| MOVE to SR | Move to the Status Register | (Source)→SR | * | * | * | * | * |
| MOVE from SR | Move from the Status Register | SR→Destination | - | - | - | - | - |
| MOVE USP | Move User Stack Pointer | USP→An; An→USP | - | - | - | - | - |
| MOVEA | Move Address | (Source)→(Destination) | - | - | - | - | - |
| MOVEM | Move Multiple Registers | Registers→Destination (Source)→Registers | - | - | - | - | - |
| MOVEP | Move Peripheral Data | (Source)→Destination | - | - | - | - | - |
| MOVEQ | Move Quick | Immediate Data→Destination | - | * | * | 0 | 0 |
| MULS | Signed Multiply | (Destination)*(Source)→Destination | - | * | * | 0 | 0 |
| MULU | Unsigned Multiply | (Destination)*(Source)→Destination | - | * | * | 0 | 0 |
| NBCD | Negate Decimal with Extend | 0-(Destination) ₁₀ -X→Destination | * | U | * | U | * |
| NEG | Negate | 0-(Destination)→Destination | * | * | * | * | * |
| NEGX | Negate with Extend | 0-(Destination) ₁₀ -X→Destination | * | * | * | * | * |
| NOP | No Operation | - | - | - | - | - | - |
| NOT | Logical complement | ~(Destination)→Destination | - | * | * | 0 | 0 |
| OR | Inclusive OR Logical | (Destination) v (Source)→Destination | - | * | * | 0 | 0 |
| ORI | Inclusive OR Immediate | (Destination) v Immediate Data→Destination | - | * | * | 0 | 0 |
| PEA | Push Effective Address | Destination→SP@- | - | - | - | - | - |
| RESET | Reset External Devices | - | - | - | - | - | - |
| ROL, ROR | Rotate (Without Extend) | (Destination) Rotated by <count>→Destination | - | * | * | 0 | * |
| ROXL, ROXR | Rotate with Extend | (Destination) Rotated by <count>→Destination | * | * | * | 0 | * |
| RTE | Return from Exception | SP@+→SR; SP@+→PC | * | * | * | * | * |
| RTR | Return and Restore Condition Codes | SP@+→CC; SP@+→PC | * | * | * | * | * |
| RTS | Return from Subroutine | SP@+→PC | - | - | - | - | - |
| SBCD | Subtract Decimal with Extend | (Destination) ₁₀ -(Source) ₁₀ -X→Destination | * | U | * | U | * |
| SCC | Set According to Condition | If CC then 1's→Destination else 0's→Destination | - | - | - | - | - |
| STOP | Load Status Register and Stop | Immediate Data→SR; STOP | * | * | * | * | * |
| SUB | Subtract Binary | (Destination)-(Source)→Destination | * | * | * | * | * |
| SUBA | Subtract Address | (Destination)-(Source)→Destination | - | - | - | - | - |
| SUBI | Subtract Immediate | (Destination)-Immediate Data→Destination | * | * | * | * | * |
| SUBQ | Subtract Quick | (Destination)-Immediate Data→Destination | * | * | * | * | * |
| SUBX | Subtract with Extend | (Destination)-(Source)-X→Destination | * | * | * | * | * |
| SWAP | Swap Register Halves | Register [31:16]↔Register [15:0] | - | * | * | 0 | 0 |

16/32-bit microprocessor**SCC68070**

| MNEMONIC | DESCRIPTION | OPERATION | CONDITION CODES | | | | |
|----------|----------------------|---|-----------------|---|---|---|---|
| | | | X | N | Z | V | C |
| TAS | Test and Set Operand | (Destination) Tested→CC, 1→[7] OF Destination | - | * | * | 0 | 0 |
| TRAP | Trap | PC→SSP@-; SR→SSP@-; (Vector)→PC | - | - | - | - | - |
| TRAPV | Trap on overflow | If V then TRAP | - | - | - | - | - |
| TST | Test an Operand | (Destination) Tested→CC | - | * | * | 0 | 0 |
| UNLK | Unlink | An→SP; SP@+→An | - | - | - | - | - |

Notes to Table 5

1. [] = bit number
2. * affected
3. - unaffected
4. 0 cleared
5. 1 set
6. U defined
7. @ location addressed by

Notes to Tables 5 to 19

EA = Effective Address
 An = Address Register
 Dn = Data Register
 Xn = Address or Data Register used as index register
 N = 1 for bytes, 2 for words and 4 for long words
 ← = Replaces
 SR = Status Register
 PC = Program Counter
 () = Contents of
 d₈ = 8-bit offset (displacement)
 d₁₆ = 16-bit offset (displacement)
 SP = Stack Pointer
 SSP = System Stack Pointer
 USP = User Stack Pointer

Table 6 Data addressing modes.

| MODE | GENERATION |
|---|---|
| Register Direct Addressing Data Register Direct Address Register Direct | EA = Dn EA = An |
| Absolute Data Addressing Absolute Short Absolute Long | EA = (Next Words) EA = (Next Two Words) |
| Program Counter Relative Addressing Relative with Offset Relative with Index and Offset | EA = (PC)+d ₁₆ EA = (PC)+(Xn)+d ₈ |
| Register Indirect Addressing Register Indirect Postincrement Register Indirect Predecrement Register Indirect Register Indirect with Offset Indexed Register Indirect with Offset | EA = (An) EA = (An), An←An+N An←An-N, EA = (An) EA = (An)+d ₁₆ EA = (An)+(Xn)+d ₈ |
| Immediate Data Addressing Immediate Quick Immediate | DATA = Next Word(s) Inherent Data |
| Implied Addressing Implied Register | EA = SR, USP, SSP, PC, SP |

16/32-bit microprocessor**SCC68070****INSTRUCTION TIMING****Table 7** Effective address calculation times.

| | ADDRESSING MODE | .B | .W | .L | |
|----------|---|----|-------|----|-------|
| Rn | Data or Address Register Direct | 0 | (0/0) | 0 | (0/0) |
| (An) | Address Register Indirect | 4 | (1/0) | 8 | (2/0) |
| (An)+ | Address Register Indirect postincrement | 4 | (1/0) | 8 | (2/0) |
| -(An) | Address Register Indirect predecrement | 7 | (1/0) | 11 | (2/0) |
| d(An) | Address Register Indirect Displacement | 11 | (2/0) | 15 | (3/0) |
| d(An,Xi) | Address Register Indirect with Index | 14 | (2/0) | 18 | (3/0) |
| xxx.S | Absolute Short | 8 | (2/0) | 12 | (3/0) |
| xxx.L | Absolute Long | 12 | (3/0) | 16 | (4/0) |
| d(PC) | Program Counter with Displacement | 11 | (2/0) | 15 | (3/0) |
| d(PC,Xi) | Program Counter with Index | 14 | (2/0) | 18 | (3/0) |
| #xxx | Immediate | 4 | (1/0) | 8 | (2/0) |

The number of bus read and write cycles are shown in parenthesis.

Note.

In Tables 8 to 19 the clock period is twice the crystal oscillator period.

Table 8 MOVE Byte and MOVE Word instruction clock periods.

| | Rn | (An) | (An)+ | -(An) | d(An) | d(An,iX) | xxx.S | xxx.L |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| Rn | 7 (1/0) | 11 (1/1) | 11 (1/1) | 14 (1/1) | 18 (2/1) | 21 (2/1) | 15 (2/1) | 19 (3/1) |
| (An) | 11 (2/0) | 15 (2/1) | 15 (2/1) | 18 (2/1) | 22 (3/1) | 25 (3/1) | 19 (3/1) | 23 (4/1) |
| (An)+ | 11 (2/0) | 15 (2/1) | 15 (2/1) | 18 (2/1) | 22 (3/1) | 25 (3/1) | 19 (3/1) | 23 (4/1) |
| -(An) | 14 (2/0) | 18 (2/1) | 18 (2/1) | 21 (2/1) | 25 (3/1) | 28 (3/1) | 22 (3/1) | 26 (4/1) |
| d(An) | 18 (3/0) | 22 (3/1) | 22 (3/1) | 25 (3/1) | 29 (4/1) | 32 (4/1) | 26 (4/1) | 30 (5/1) |
| d(An,Xi) | 21 (3/0) | 25 (3/1) | 25 (3/1) | 28 (3/1) | 32 (4/1) | 35 (4/1) | 29 (4/1) | 33 (5/1) |
| xxx.S | 15 (3/0) | 19 (3/1) | 19 (3/1) | 22 (3/1) | 26 (4/1) | 29 (4/1) | 23 (4/1) | 27 (5/1) |
| xxx.L | 19 (4/0) | 23 (4/1) | 23 (4/1) | 26 (4/1) | 30 (5/1) | 33 (5/1) | 27 (5/1) | 31 (6/1) |
| d(PC) | 18 (3/0) | 22 (3/1) | 22 (3/1) | 25 (3/1) | 29 (4/1) | 32 (4/1) | 26 (4/1) | 30 (5/1) |
| d(PC,Xi) | 21 (3/0) | 25 (3/1) | 25 (3/1) | 28 (3/1) | 32 (4/1) | 35 (4/1) | 29 (4/1) | 33 (5/1) |
| #xxx | 11 (2/0) | 15 (2/1) | 15 (2/1) | 18 (2/1) | 22 (3/1) | 25 (3/1) | 19 (3/1) | 23 (4/1) |

Table 9 MOVE long instruction clock periods.

| | Rn | (An) | (An)+ | -(An) | d(An) | d(An,iX) | xxx.S | xxx.L |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| Rn | 7 (1/0) | 15 (1/2) | 15 (1/2) | 18 (1/2) | 22 (2/2) | 25 (2/2) | 19 (2/2) | 23 (3/2) |
| (An) | 15 (3/0) | 23 (3/2) | 23 (3/2) | 26 (3/2) | 30 (4/2) | 33 (4/2) | 27 (4/2) | 31 (5/2) |
| (An)+ | 15 (3/0) | 23 (3/2) | 23 (3/2) | 26 (3/2) | 30 (4/2) | 33 (4/2) | 27 (4/2) | 31 (5/2) |
| -(An) | 18 (3/0) | 26 (3/2) | 26 (3/2) | 29 (3/2) | 33 (4/2) | 36 (4/2) | 30 (4/2) | 34 (5/2) |
| d(An) | 22 (4/0) | 30 (4/2) | 30 (4/2) | 33 (4/2) | 37 (5/2) | 40 (5/2) | 34 (5/2) | 38 (6/2) |
| d(An,Xi) | 25 (4/0) | 33 (4/2) | 33 (4/2) | 36 (4/2) | 40 (5/2) | 43 (5/2) | 37 (5/2) | 41 (6/2) |
| xxx.S | 19 (4/0) | 27 (4/2) | 27 (4/2) | 30 (4/2) | 34 (5/2) | 37 (5/2) | 31 (5/2) | 35 (6/2) |
| xxx.L | 23 (5/0) | 31 (5/2) | 31 (5/2) | 34 (5/2) | 38 (6/2) | 41 (6/2) | 35 (6/2) | 39 (7/2) |
| d(PC) | 22 (4/0) | 30 (4/2) | 30 (4/2) | 33 (4/2) | 37 (5/2) | 40 (5/2) | 34 (5/2) | 38 (6/2) |
| d(PC,Xi) | 25 (4/0) | 33 (4/2) | 33 (4/2) | 36 (4/2) | 40 (5/2) | 43 (5/2) | 37 (5/2) | 41 (6/2) |
| #xxx | 15 (3/0) | 23 (3/2) | 23 (3/2) | 26 (3/2) | 30 (4/2) | 33 (4/2) | 27 (4/2) | 31 (5/2) |

16/32-bit microprocessor**SCC68070****Table 10** Standard instruction clock periods.

| | SIZE | | op <ea>,An | | op <ea>,Dn | | op Dn,<M> | |
|----------|------|----|------------|-------|--------------------|----------------------|-----------|-------|
| ADD/ADDA | .B | .W | 7+ | (1/0) | 7+ | (1/0) | 11+ | (1/1) |
| | .L | | 7+ | (1/0) | 7+ | (1/0) | 15+ | (1/2) |
| ADD | .B | .W | - | | 7+ | (1/0) | 11+ | (1/1) |
| | .L | | - | | 7+ | (1/0) | 15+ | (1/2) |
| CMP/CMPA | .B | .W | 7+ | (1/0) | 7+ | (1/0) | - | |
| | .L | | 7+ | (1/0) | 7+ | (1/0) | - | |
| DIVS | - | | - | | 169+ ^{**} | (1/0) | - | |
| DIVU | - | | - | | 130+ [*] | (1/0) | - | |
| EOR | .B | .W | - | | 7+ | (1/0) ^{***} | 11+ | (1/1) |
| | .L | | - | | 7+ | (1/0) ^{***} | 15+ | (1/2) |
| MULS | - | | - | | 76+ [*] | (1/0) | - | |
| MULU | - | | - | | 76+ [*] | (1/0) | - | |
| OR | .B | .W | - | | 7+ | (1/0) | 11+ | (1/1) |
| | .L | | - | | 7+ | (1/0) | 15+ | (1/2) |
| SUB | .B | .W | 7+ | (1/0) | 7+ | (1/0) | 11+ | (1/1) |
| | .L | | 7+ | (1/0) | 7+ | (1/0) | 15+ | (1/2) |

+ add effective address calculation time

* the duration of the instruction is constant

** indicates maximum value

*** only effective address mode is data register direct

Table 11 Immediate instruction clock periods.

| | SIZE | | op <#>,Dn | | op <#>,An | | op <#>,<M> | |
|-------|------|----|-----------|-------|-----------|-------|------------|-------|
| ADDI | .B | .W | 14 | (2/0) | - | | 18+ | (2/1) |
| | .L | | 18 | (3/0) | - | | 26+ | (3/2) |
| ADDQ | .B | .W | 7 | (1/0) | 7 | (1/0) | 11+ | (1/1) |
| | .L | | 7 | (1/0) | 7 | (1/0) | 15+ | (1/2) |
| ANDI | .B | .W | 14 | (2/0) | - | | 18+ | (2/1) |
| | .L | | 18 | (3/0) | - | | 26+ | (3/2) |
| CMPI | .B | .W | 14 | (2/0) | - | | 14+ | (2/0) |
| | .L | | 18 | (3/0) | - | | 18+ | (3/0) |
| EORI | .B | .W | 14 | (2/0) | - | | 18+ | (2/1) |
| | .L | | 18 | (3/0) | - | | 26+ | (3/2) |
| MOVEQ | .L | | 7 | (1/0) | - | | - | |
| ORI | .B | .W | 14 | (2/0) | - | | 18+ | (2/1) |
| | .L | | 18 | (3/0) | - | | 26+ | (3/2) |
| SUBI | .B | .W | 14 | (2/0) | - | | 18+ | (2/1) |
| | .L | | 18 | (3/0) | - | | 26+ | (3/2) |
| SUBQ | .B | .W | 7 | (1/0) | 7 | (1/0) | 11+ | (1/1) |
| | .L | | 7 | (1/0) | 7 | (1/0) | 15+ | (1/2) |

+ add effective address calculation time

16/32-bit microprocessor**SCC68070****Table 12** Single operands instruction clock periods.

| INSTRUCTION | SIZE | REGISTER | | MEMORY | |
|-------------|------------|----------|-------|--------|----------|
| | | | | | |
| CLR | Byte, Word | 7 | (1/0) | 11 | (1/1)+* |
| | Long | 7 | (1/0) | 15 | (1/2)+** |
| NBCD | Byte | 10 | (1/0) | 14 | (1/1)+ |
| NEG | Byte, Word | 7 | (1/0) | 11 | (1/1)+ |
| | Long | 7 | (1/0) | 15 | (1/2)+ |
| NEGX | Byte, Word | 7 | (1/0) | 11 | (1/2)+ |
| | Long | 7 | (1/0) | 15 | (1/2)+ |
| NOT | Byte, Word | 7 | (1/0) | 11 | (1/1)+ |
| | Long | 7 | (1/0) | 15 | (1/2)+ |
| Scc | Byte, Word | 13 | (1/0) | 17 | (1/1)+ |
| | Long | 13 | (1/0) | 14 | (1/1)+ |
| TAS | Byte | 10 | (1/0) | 15 | (1/1)+* |
| TST | Byte, Word | 7 | (1/0) | 7 | (1/0)+ |
| | Long | 7 | (1/0) | 7 | (1/0)+ |

+ add effective address calculation time

* subtract one read cycle (-4(1/0)) from effective address calculation

** subtract two read cycles (-8(2/0)) from effective address calculation

Table 13 Shift/rotate instruction clock periods.

| INSTRUCTION | SIZE | REGISTER | | MEMORY | |
|-------------|------------|----------|-------|--------|--------|
| | | | | | |
| ASR, ASL | Byte, Word | 13 + 3n | (1/0) | 14 | (1/1)+ |
| | Long | 13 + 3n | (1/0) | - | |
| LSR, LRL | Byte, Word | 13 + 3n | (1/0) | 14 | (1/1)+ |
| | Long | 13 + 3n | (1/0) | - | |
| ROR, ROL | Byte, Word | 13 + 3n | (1/0) | 14 | (1/1)+ |
| | Long | 13 + 3n | (1/0) | - | |
| ROXR, ROXL | Byte, Word | 13 + 3n | (1/0) | 14 | (1/1)+ |
| | Long | 13 + 3n | (1/0) | - | |

+ add effective address calculation time

Table 14 Bit manipulation instruction clock periods.

| INSTRUCTION | SIZE | DYNAMIC | | STATIC | |
|-------------|------|----------|-----------|----------|-----------|
| | | REGISTER | MEMORY | REGISTER | MEMORY |
| BCHG | Byte | - | 14 (1/1)+ | - | 21 (2/1)+ |
| | Long | 10 (1/0) | - | 17 (2/0) | - |
| BCLR | Byte | - | 14 (1/1)+ | - | 21 (2/1)+ |
| | Long | 10 (1/0) | - | 17 (2/0) | - |
| BSET | Byte | - | 14 (1/1)+ | - | 21 (2/1)+ |
| | Long | 10 (1/0) | - | 17 (2/0) | - |
| BTST | Byte | - | 7 (1/0)+ | - | 14 (2/0)+ |
| | Long | 7 (1/0) | - | 14 (2/0) | - |

+ add effective address calculation time

16/32-bit microprocessor

SCC68070

Table 15 Conditional instruction clock periods.

| INSTRUCTION | DISPL. | TRAP/ BRANCH TAKEN | TRAP/ BRANCH NOT TAKEN |
|-------------|---------|--------------------------|------------------------------|
| Bcc | .B | 13 (1/0) | 13 (1/0) |
| | .W | 14 (2/0) | 14 (2/0) |
| BRA | .B | 13 (1/0) | - |
| | .W | 14 (2/0) | - |
| BSR | .B | 17 (1/2) | - |
| | .W | 22 (2/2) | - |
| DBcc | ccTrue | - | 14 (2/0) |
| | ccFalse | 17 (2/0) | 17 (2/0) |
| CHK | - | 64 (3/5)+ | 19 (1/0)+ |
| TRAPV | - | 55 (3/4) | 10 (1/0) |

+ add effective address calculation time

Table 16 JMP, JSR, LEA, PEA, MOVEM instruction clock periods.

| INSTR. | SIZE | (An) | (An)+ | -(An) | d(An) | d(An,Xi) | xxx.S | xxx.L | d(PC) | d(PC,Xi) |
|--------|------|--------------------|--------------------|------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| JMP | - | 7 | - | - | 14 | 17 | 14 | 18 | 14 | 17 |
| | - | (1/0) | - | - | (2/0) | (2/0) | (2/0) | (3/0) | (2/0) | (2/0) |
| JSR | - | 18 | - | - | 25 | 28 | 25 | 29 | 25 | 28 |
| | - | (1/2) | - | - | (2/2) | (2/2) | (2/2) | (3/2) | (2/2) | (2/2) |
| LEA | - | 7 | - | - | 14 | 17 | 14 | 18 | 14 | 17 |
| | - | (1/0) | - | - | (2/0) | (2/0) | (2/0) | (3/0) | (2/0) | (2/0) |
| PEA | - | 18 | - | - | 25 | 28 | 25 | 29 | 25 | 28 |
| | - | (1/2) | - | - | (2/2) | (2/2) | (2/2) | (3/2) | (2/2) | (2/2) |
| MOVEM | .W | 26+7n (2+n/0) | 26+7n (2+n/0) | - | 30+7n (3+n/0) | 33+7n (3+n/0) | 30+7n (3+n/0) | 34+7n (4+n/0) | 30+7n (3+n/0) | 33+7n (3+n/0) |
| M»R | .L | 26+11n (2+2n/0) | 26+11n (2+2n/0) | - | 30+11n (3+2n/0) | 33+11n (3+2n/0) | 30+11n (3+2n/0) | 34+11n (4+2n/0) | 30+11n (3+2n/0) | 33+11n (3+2n/0) |
| MOVEM | .W | 23+7n (2/n) | - | 23+7n (2/n) | 27+7n (3/n) | 30+7n (3/n) | 27+7n (3/n) | 31+7n (4/n) | - | - |
| R»M | .L | 23+11n (2/2n) | - | 23+11n (2/2n) | 27+11n (3/2n) | 30+11n (3/2n) | 27+11n (3/2n) | 31+11n (4/2n) | - | - |

n = number of registers to move

Table 17 Multi-precision instruction clock periods.

| INSTRUCTION | SIZE | op Dn, Dn | op M, M |
|-------------|------------|-----------|----------|
| ADDX | Byte, Word | 7 (1/0) | 28 (3/1) |
| | Long | 7 (1/0) | 40 (5/2) |
| CMPM | Byte, Word | - | 18 (3/0) |
| | Long | - | 26 (5/0) |
| SUBX | Byte, Word | 7 (1/0) | 28 (3/1) |
| | Long | 7 (1/0) | 40 (5/2) |
| ABCD | Byte | 10 (1/0) | 31 (3/1) |
| SBCD | Byte | 10 (1/0) | 31 (3/1) |

16/32-bit microprocessor**SCC68070****Table 18** Miscellaneous clock periods.

| INSTRUCTION | SIZE | REGISTER | MEMORY | REGISTER TO MEMORY | MEMORY TO REGISTER |
|---------------|------|------------|-----------|--------------------|--------------------|
| ANDI to CCR | - | 14 (2/0) | - | - | - |
| ANDI to SR | - | 14 (2/0) | - | - | - |
| EORI to CCR | - | 14 (2/0) | - | - | - |
| EORI to SR | - | 14 (2/0) | - | - | - |
| EXG | - | 13 (1/0) | - | - | - |
| EXT | Word | 7 (1/0) | - | - | - |
| | Long | 7 (1/0) | - | - | - |
| LINK | - | 25 (2/2) | - | - | - |
| MOVE from SR | - | 7 (1/0) | 11/11+ | - | - |
| MOVE to CCR | - | 10 (2/0) | 10 (2/0)+ | - | - |
| MOVE to SR | - | 10 (2/0) | 10 (2/0)+ | - | - |
| MOVE from USP | - | 7 (1/0) | - | - | - |
| MOVE to USP | - | 7 (1/0) | - | - | - |
| MOVEP | Word | - | - | 25 (2/2) | 22 (4/0) |
| | Long | - | - | 39 (2/4) | 36 (6/0) |
| NOP | - | 7 (1/0) | - | - | - |
| ORI to CCR | - | 14 (2/0) | - | - | - |
| ORI to SR | - | 14 (2/0) | - | - | - |
| RESET | - | 154 (2/0) | - | - | - |
| RTE | | | | | |
| short format | | 39 (5/0) | | | |
| long format: | | | | | |
| no rerun | | 140 (18/0) | | | |
| with rerun | | 146 (18/0) | | | |
| rerun of TAS | | 151 (19/1) | | | |
| RTR | - | 22 (4/0) | - | - | - |
| RTS | - | 15 (3/0) | - | - | - |
| STOP | - | 13 (0/0) | - | - | - |
| SWAP | - | 7 (1/0) | - | - | - |
| UNLK | - | 15 (3/0) | - | - | - |

+ add effective address calculation time

Table 19 Exception processing clock periods.

| EXCEPTION | NO. OF CLOCK PERIODS |
|---------------------|----------------------|
| Address error | 158 (3/17) |
| Bus error | 158 (3/17) |
| Interrupt | 65 (4/4)* |
| Illegal instruction | 55 (3/4) |
| Privilege violation | 55 (3/4) |
| Trace | 55 (3/4) |
| Trap | 52 (3/4) |
| Divide by Zero | 64 (3/4)+ |
| RESET** | 43 (4/0) |

* The interrupt acknowledge bus cycle is assumed to take four external clock periods.

** Indicates the maximum time from when RESETN and HALTN are first sampled as negated to first instruction fetch.

16/32-bit microprocessor**SCC68070****ON-CHIP MMU**

The SCC68070 has an on-chip Memory Management Unit (MMU) that if enabled, supports virtual memory, multi-tasking, task protection and dynamic stack allocation.

Segmentation

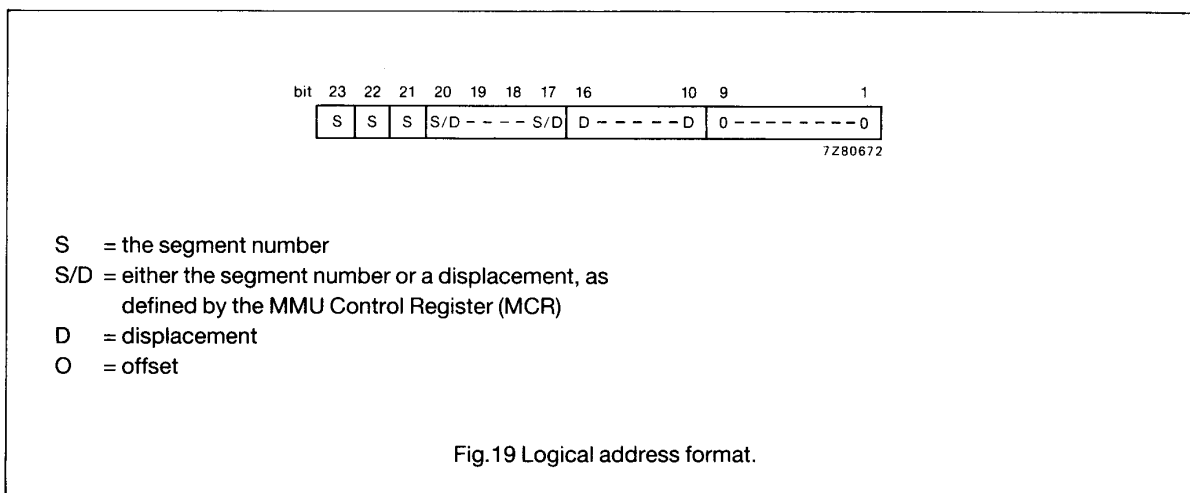
The MMU divides the memory into segments of multiples of 1 Kbyte (blocks). Two modes are possible:

| MODE | NO. OF SEGMENTS | MAX. SEGMENT LENGTH |
|------|-----------------|-------------------------|
| 1 | 8 | 2048 blocks = 2 Mbytes |
| 2 | 128 | 128 blocks = 128 Kbytes |

Memory protection is assigned on a segment to segment basis.

For address translation, the logical address can be split into three parts (see Fig. 19).

- segment number
- displacement
- offset

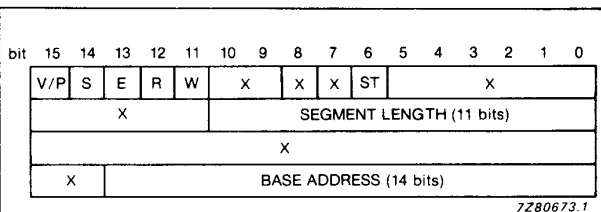


16/32-bit microprocessor

SCC68070

Segment descriptors

Every segment is described by a segment descriptor stored in the segment descriptor table in main memory. Each descriptor contains a segment address field, a segment protection field (with protection attributes) and the segment length. However, for ease of access up to 8 descriptors can be stored in the on-chip descriptor RAM. The format of the descriptors in main memory is shown in Fig.20.



X = undefined, reserved

- Base Address: the 14 MSBs of the start address of this segment.
- V/P (Valid/Present) - may be used by the loading software to determine whether the segment is valid and present.
- S (Supervisor) - supervisor permission is required to access this segment.
- E (Execute) - instruction fetches may be performed with this segment.
- R (Read) - read operations may be performed with this segment.
- W (Write) - write operations may be performed with this segment.
- ST (Stack) - stack segments grow from high to low addresses.

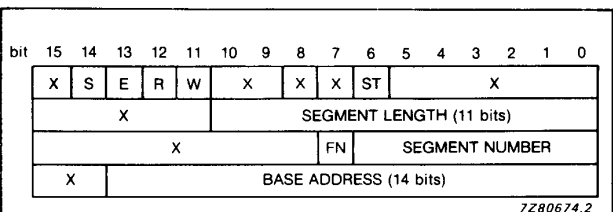
Segment length: the 11 bits define the length of a segment by the number of 1 Kbyte blocks. (If the SN bit of the MMU Control Register (MCR) is set, placing the MMU in Mode 2, only 7 MSBs define the length of the segment.)

Note:

All bits indicated as "undefined, reserved" must be programmed to zeros, except bit 8 of the first word which must be set to '1' to be compatible with the Memory Access Controllers of the 68000 family.

Fig.20 Format of descriptors in main memory.

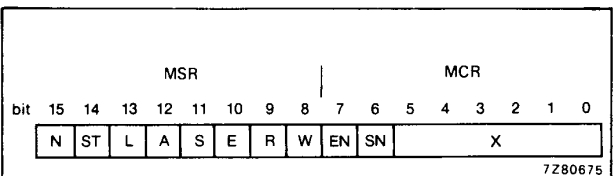
The format of the descriptor in the on-chip descriptor RAM is shown in Fig.21. The low-order byte of the third word contains the segment number that is loaded into the fully associative CAM. The MSB of the segment number is in Bit 6 of this word, i.e. it is left justified for either mode of the segment number.



FN = FLUSH If this bit is equal to zero it invalidates the descriptor. Therefore, it must be reset to zero when writing a new valid segment number. The FN bit should be set to one after the remainder of the segment descriptor has been loaded into the MMU.

X Undefined, reserved; will return zeros when ready by the CPU.

Fig.21 Descriptor format in MMU.



X = undefined, reserved

Fig.22 MMU Status and Control Registers (MSR and MCR).

16/32-bit microprocessor**SCC68070****MMU Control and Status Registers**

The formats of the Control Register (MCR) and the Status Register (MSR) are shown in Fig.22. These registers can be accessed as either one word or two separate bytes. MCR may be written to, or read from, but MSR is a read only register. Writing to MSR will result in a normal bus cycle with no effect.

Control Register (MCR)**ENABLE (MCR 7)**

If the Enable bit EN is set the MMU is enabled (address translation and protection). The default reset value is zero, inhibiting the MMU.

NUMBER OF SEGMENTS (MCR 6)

The state of the SN bit determines the maximum number of segments the MMU is to handle. When SN is '0', the maximum number of segments is 8 (Mode 1). When SN is '1', the maximum number of segments is 128 (Mode 2). The default reset value is zero.

Status Register (MSR)**NOT-PRESENT (MSR 15)**

The Not-present bit N, is set by the MMU if the addressed segment has no descriptor in the MMU or when the FN bit in the descriptor is zero.

STACK SEGMENT (MSR 14)

The Stack segment bit ST, indicates that the segment in which the error occurred was defined as a stack segment. If ST is set, the segment grows from the highest to lowest address.

LENGTH VIOLATION (MSR 13)

The Length violation bit L, is set if the displacement given by the logical address exceeds the segment length given by the segment descriptor.

ACCESS ERROR (MSR 12)

The Access error bit A, is set by the MMU if an attribute violation occurred during the last bus access.

DESCRIPTOR ATTRIBUTES (MSR 11:8)

The Supervisor bit S (MSR 11), Execute bit E (MSR 10), Read bit R (MSR 9) and Write bit W (MSR 8) are permission bits and reflect the descriptor attributes when an access error has occurred.

Address map of the MMU

The internal addresses of the MMU's registers and Descriptor RAM are given in below:

Base address 8000 8000 (HEX)

| A6 | A5 | A4 | A3 | A2 | A1 | A0 | |
|----|----|----|----|----|----|-----|---------------------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | MSR |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | MCR |
| 1 | C | C | C | 0 | 0 | 0/1 | ATTR |
| 1 | C | C | C | 0 | 1 | 0/1 | SEG LENGTH |
| 1 | C | C | C | 1 | 0 | 0 | Undefined, reserved |
| 1 | C | C | C | 1 | 0 | 1 | SEG NUMBER |
| 1 | C | C | C | 1 | 1 | 0/1 | BASE ADDRESS |

Notes

MSR = MMU Status Register

MCR = MMU Control Register

SEG = Segment

ATTR = Attributes

CCC = 000 to 111 (see Segment Descriptors 0 to 7).

All internal MMU addresses are mapped in the SCC68070's on-chip address space and are only accessible in Supervisor mode (see Tables 1 and 2). On-chip addresses are not processed by the MMU.

Operation

The MMU provides protection and virtual memory support. When the MMU is enabled it only influences addresses sent by the CPU and adds two SM states (or one clock cycle) at the beginning of each external bus cycle of the CPU. However, during the vector acquisition of an interrupt acknowledge cycle, the MMU does not process the address coming from the CPU; i.e. logical address equals physical address and the SM states are not added. The address translation is processed as follows:

16/32-bit microprocessor

SCC68070

1. The segment number S, of the logical address is used to address the segment descriptor stored in the MMU on-chip descriptor RAM. If this segment descriptor is not valid or not present, BERRN will be generated and corrective action must be taken to load the indicated segment descriptor and continue the interrupted instruction.
2. The displacement D, of the logical address is compared to the length attribute in the segment descriptor. If the displacement is outside the indicated length then BERRN is generated.
3. The internal function codes are used to check for access violations, using the attributes of the segment descriptor. In the event of a violation BERRN will be generated.
4. If no violations have been detected, a physical address will be generated. This address is constructed by adding the segment base address, the displacement D and the offset '0' bits. This is illustrated in Fig.23.

Error handling.

If an error occurs within the enabled MMU, the MMU inhibits data strobes (UDSN, LDSN) and asserts a bus error signal (BERRN) to both the CPU and the external world. The address strobe (ASN) will be asserted to indicate bus occupation to other possible bus masters. Then, the CPU will start bus error processing with the stack operation.

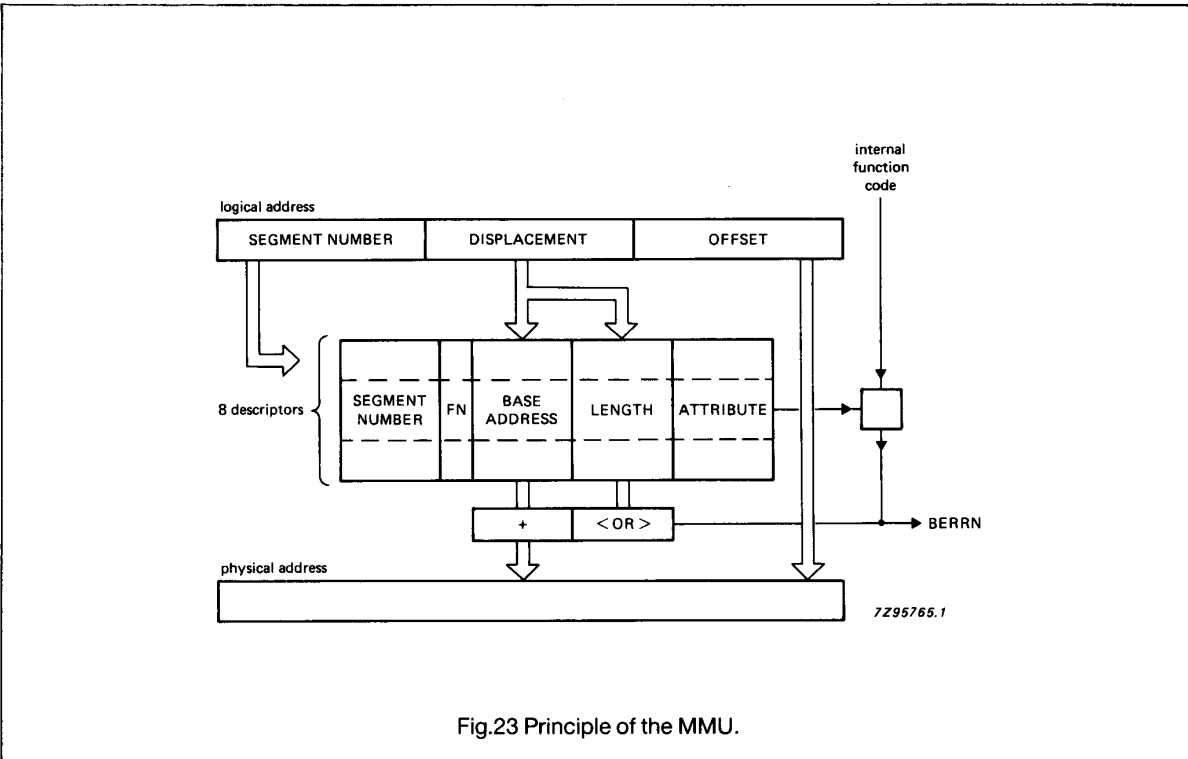


Fig.23 Principle of the MMU.

16/32-bit microprocessor

SCC68070

DMA CONTROLLER

General

The SCC68070 has on-chip, a two-channel DMA controller that handles byte or word operands, and devices with port sizes of 8 or 16-bits. The channels can be programmed to transfer data in cycle steal (single cycle) or burst (a block of successive cycles) mode. Channel 1 always uses single addressing, while Channel 2 can operate with single or dual addresses (memory to memory).

Typical system latency times are less than 1.7 μ s and the maximum transfer rate (using single addressing) is 2.98 million transfers per second (crystal frequency 35 MHz). The SCC68070's DMA controller is a subset of the existing DMA controllers in the 68000 family (68430, 68440 and 68450) and is therefore programmed similar to these devices.

Device/DMA controller communication

The following five signal lines enable peripheral devices and the DMA controller to communicate with each other. See pin description for further details.

- Request (REQ1N and REQ2N). The device makes a request for service by asserting the REQ1N (or REQ2N) line. Either burst or cycle steal mode can be used.
- ACKNOWLEDGE (ACK1N and ACK2N). The channel asserts the acknowledge line (which implicitly addresses the device making the request) during transfers to and from the device.
- READY (RDYN). RDYN is an active-LOW input which is asserted by the requesting device in single-address mode.
- Device Transfer Complete (DTCN). DTCN is an active LOW output asserted by the DMA controller during device bus cycles to indicate that the cycle has been completed successfully.
- Done (DONEN). DONEN is a bidirectional active-LOW signal. As an output it indicates to the device that the memory transfer count is exhausted. As an input, it indicates that the operation will terminate after current operand has been transferred. (See Termination phase).

Bus arbitration and priority resolution

The SCC68070 contains three possible bus masters, the DMA Channels 1 and 2 and the CPU, where Channel 1 has the highest priority and the CPU the lowest. There can also be external bus masters which have a priority lower than Channel 2 but higher than the CPU (which has the lowest priority in the system).

When a valid bus transfer request is received from a device, the DMA controller will arbitrate for and acquire the bus. The DMA controller indicates to the CPU that it wants to become bus master by generating an internal bus request signal, and the CPU responds by sending an internal bus grant signal, daisy-chained through the DMA controller. It is not offered to the external devices because the DMA controller has a request pending. If BGN has already been offered to the external devices when a DMA request arrives, the DMA controller waits for BGACKN to become inactive.

If the CPU is the current bus master, it will finish its current cycle and then give the bus to the DMA controller. When the DMA controller has received the internal bus grant signal, it waits for the external signals, Address Strobe (ASN) and Data Transfer Acknowledge (DACKN) to become inactive, it then assumes bus ownership. If the BRN signal is asserted by an external device when the CPU is not the bus master (i.e. BGACKN is asserted) no BGN signal will be sent until the CPU has regained the bus (i.e. BGACKN is inactive.)

Registers and counters

The internal organization of the DMA controllers accessible registers is shown in Table 20. The following rules apply to all registers:

- A read from a reserved location in the map results in a read from the 'null register'. The null register register returns all ones for data and results in a normal bus cycle. A write to one of these locations results in a normal bus cycle but no write occurs.
- Unused bits of a defined register are read as indicated in the register description.
- All registers are addressable as 8-bit quantities. Addresses are arranged so that certain sets of registers may also be accessed as either words or long words.

Compatibility with other 68000 family DMA controllers is provided by mapping control and status bits into bit positions which are equivalent to the register map of the other devices. Bits which are used in other devices but not in the DMA controller are assigned default values. If upward compatibility with the other devices is required, the programmer should use these default values when writing the control words to the registers; although they have no effect in the DMA controller. When a register is read, the default value is returned regardless of the value used when the register was programmed. The default value is indicated by '(x)' in unused bit positions in the register descriptions that follow.

16/32-bit microprocessor**SCC68070****Table 20** DMA controller address map.

Base address 8000 4000 (HEX)

| ADDRESS BITS ^(1),2) | | | | | | | | ACRONYM | REGISTER NAME | MODE | AFFECTED BY RESET |
|--------------------------------|---|---|---|---|---|---|---|---------|------------------------------------|-----------------------|-------------------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | |
| c | c | 0 | 0 | 0 | 0 | 0 | 0 | CSR | Channel Status Register | R/W ³⁾ | Yes |
| c | c | 0 | 0 | 0 | 0 | 0 | 1 | CER | Channel Error Register | R | Yes |
| c | c | 0 | 0 | 0 | 0 | 1 | 0 | | Reserved | | |
| c | c | 0 | 0 | 0 | 0 | 1 | 1 | | Reserved | | |
| c | c | 0 | 0 | 0 | 1 | 0 | 0 | DCR | Device Control Register | R/W | Yes |
| c | c | 0 | 0 | 0 | 1 | 0 | 1 | OCR | Operation Control Register | R/W | Yes |
| c | c | 0 | 0 | 0 | 1 | 1 | 0 | SCR | Sequence Control Register | R/W ⁴⁾ | Yes |
| c | c | 0 | 0 | 0 | 1 | 1 | 1 | CCR | Channel Control Register | R/W | Yes |
| c | c | 0 | 0 | 1 | 0 | 0 | 0 | | Reserved | | |
| c | c | 0 | 0 | 1 | 0 | 0 | 1 | | Reserved | | |
| c | c | 0 | 0 | 1 | 0 | 1 | 0 | MTCH | Memory Transfer Counter High | R/W | No |
| c | c | 0 | 0 | 1 | 0 | 1 | 1 | MTCL | Memory Transfer Counter Low | R/W | No |
| c | c | 0 | 0 | 1 | 1 | 0 | 0 | MACH | Memory Address Counter High | R/W ⁴⁾ | No |
| c | c | 0 | 0 | 1 | 1 | 0 | 1 | MACMH | Memory Address Counter Middle High | R/W | No |
| c | c | 0 | 0 | 1 | 1 | 1 | 0 | MACML | Memory Address Counter Middle Low | R/W | No |
| c | c | 0 | 0 | 1 | 1 | 1 | 1 | MACL | Memory Address Counter Low | R/W | No |
| c | c | 0 | 1 | 0 | 0 | d | d | | Reserved | | |
| c | c | 0 | 1 | 0 | 1 | 0 | 0 | DACH | Device Address Counter High | R/W ^{4), 5)} | No |
| c | c | 0 | 1 | 0 | 1 | 0 | 1 | DACHMH | Device Address Counter Middle High | R/W ⁵⁾ | No |
| c | c | 0 | 1 | 0 | 1 | 1 | 0 | DACML | Device Address Counter Middle Low | R/W ⁵⁾ | No |
| c | c | 0 | 1 | 0 | 1 | 1 | 1 | DACL | Device Address Counter Low | R/W ⁵⁾ | No |
| c | c | 0 | 1 | 1 | d | d | d | | Reserved | | |
| c | c | 1 | 0 | 0 | d | d | d | | Reserved | | |
| c | c | 1 | 0 | 1 | 0 | d | d | | Reserved | | |
| c | c | 1 | 0 | 1 | 1 | 0 | 0 | | Reserved | | |
| c | c | 1 | 0 | 1 | 1 | 0 | 1 | CPR | Channel Priority Register | R/W ⁴⁾ | No |
| c | c | 1 | 0 | 1 | 1 | 1 | 0 | | Reserved | | |
| c | c | 1 | 0 | 1 | 1 | 1 | 1 | | Reserved | | |
| c | c | 1 | 1 | d | d | d | d | | Reserved | | |

Notes to Table 20

- 'cc' = 00 for channel 1, 'cc' = 01 for channel 2, 'cc' = 10 or 11 reserved.
- 'd' designates don't care states.
- A write to this register may perform a status reset operation.
- This is a dummy register present only to provide compatibility with other 68000 family DMA controllers. A write to this register has no effect on the DMA controller.
- Channel 2 only.

16/32-bit microprocessor**SCC68070****Device Control Register (DCR)**

EXTERNAL REQUEST MODE (DCR 15)

This bit selects whether the channel operates in either burst or cycle steal mode, as follows:

- 0 Burst Mode: This mode allows a device to request the transfer of multiple operands over consecutive bus cycles.
- 1 Cycle steal mode: This mode allows a device to transfer operands on a per cycle basis.

DEVICE TYPE (DCR13:12) - Channel 1

These bits determine how a device is addressed, as follows:

- 11 The device connected is implicitly addressed by the 5 device control signals with handshake using ACK1N (or ACK2N) and RDYN.

DEVICE TYPE (DCR13:12) - Channel 2.

These bits determine how a device is addressed, as follows:

- 00 The device is explicitly addressed by the Device Address Counter (DAC) via the SCC68070 bus interface. Transfers are made in two bus cycles; the data being stored in the CPU between cycles.
- 11 The device connected is implicitly addressed by the 5 device control signals with handshake using ACK1N (or ACK2N) and RDYN.

DEVICE SIZE (DCR 11) - Channel 2 only

This bit functions only with explicitly addressed devices, DCR13:12 = 00

- 0 The device port size is 8-bit. The device reads/writes using only the low-order half of the bus (D0 to D7) or the high-order half (D8 to D15) depending on bit A0 of the Device Address Counter (DAC). Depending on which part of the bus is used either LDSN for the low-order part or UDSN for the high-order part is asserted. During byte size transfers (OCR 5:4 = 00) the half of the

data bus used for read/write to the memory depends on the state of A0 of the Memory Address Counter (MAC). During word size transfers (OCR 5:4 = 01), read/write operations take place in two successive bytes; the DAC will either be unchanged or incremented by two per byte, depending on SCR 9:8 (see Fig.26)

- 1 The device port size is 16-bits and the device is accessed as a normal memory location.

Operation Control Register (OCR)

DIRECTION (OCR 7)

- 0 Transfer is from memory to device.
- 1 Transfer is from device to memory.

OPERAND SIZE (OCR 5:4)

These bits determine whether UDSN, LDSN or both are generated during the transfer cycle and also to what value the address counters MAC and DAC (AC) are incremented by each transfer cycle.

16/32-bit microprocessor

SCC68070

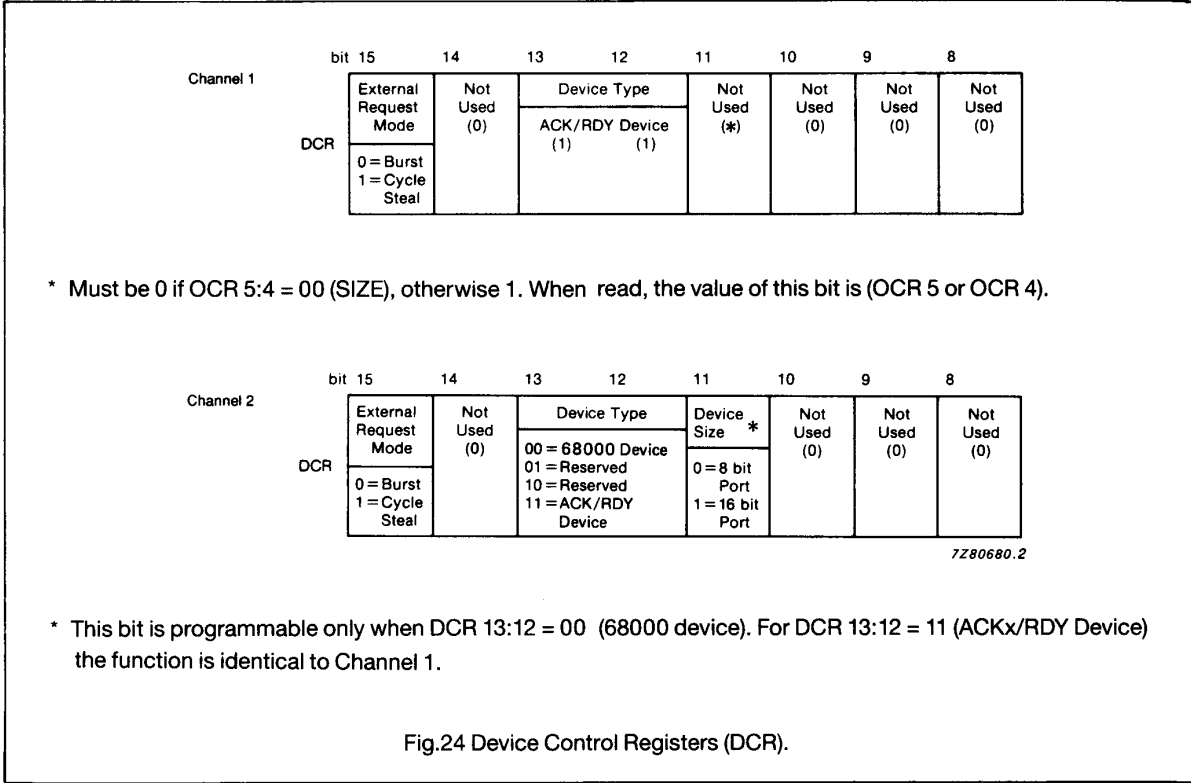


Fig.24 Device Control Registers (DCR).

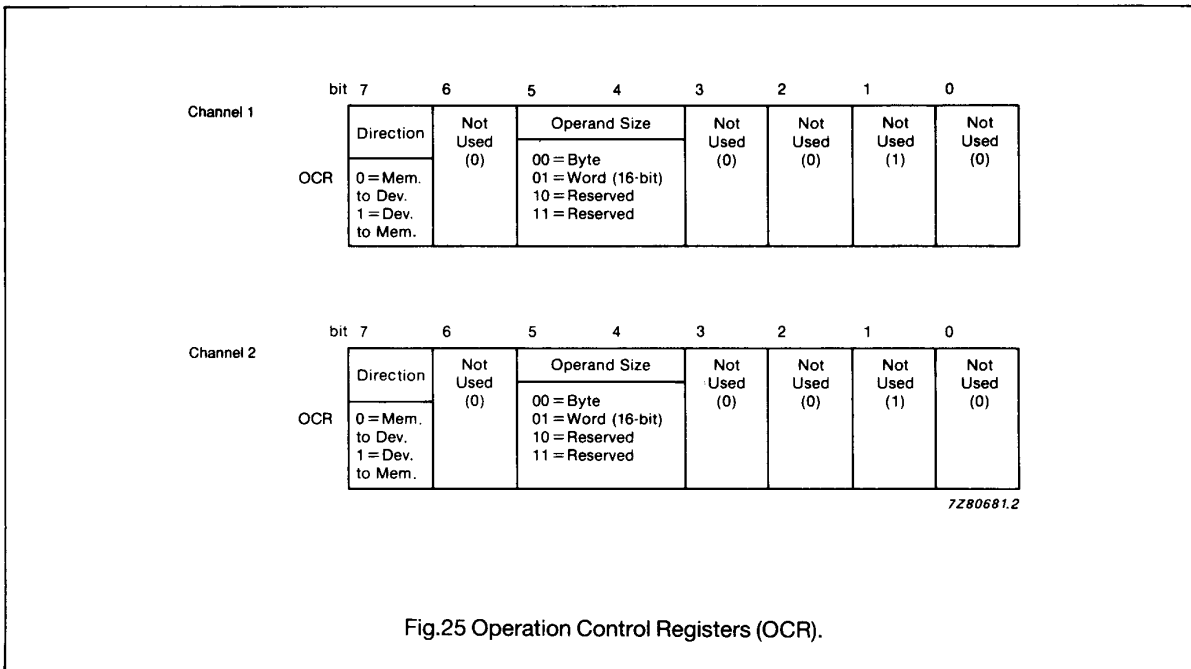


Fig.25 Operation Control Registers (OCR).

16/32-bit microprocessor

SCC68070

Sequence Control Register

MAC (SCR 11:10) - Channel 1

- 01 These bits are not programmable, the Memory Address Counter always counts upwards.

MAC (SCR 11:10) - Channel 2

- 00 The Memory Address Counter is not changed during transfers.
- 01 The Memory Address Counter is incremented by 1 or 2 during the transfer, depending on the operand size; OCR 5:4 (byte or word transfer).

DAC (SCR 9:8) - Channel 2 only

- 00 The Device Address Counter is not changed between transfers.
- 01 The Device Address Counter is incremented by 1 or 2 during the transfer, depending on the operand size; OCR 5:4 (byte or word transfer).

Channel Control Register (CCR)

START OPERATION (CCR 7)

- 0 No Start pending
- 1 Start Operation. The start bit is set to initiate operation of the channel.

SOFTWARE ABORT (CCR 4)

- 0 Do not abort.
- 1 Abort Operation. Setting this bit terminates the current operation and puts it into the Idle state, then:
 - The COC bit (CSR 15) and ERR bit (CSR 12) are set
 - The Channel Active bit (CSR 11) is reset
 - An Abort Error condition is signalled in the Channel Error Register (CER).

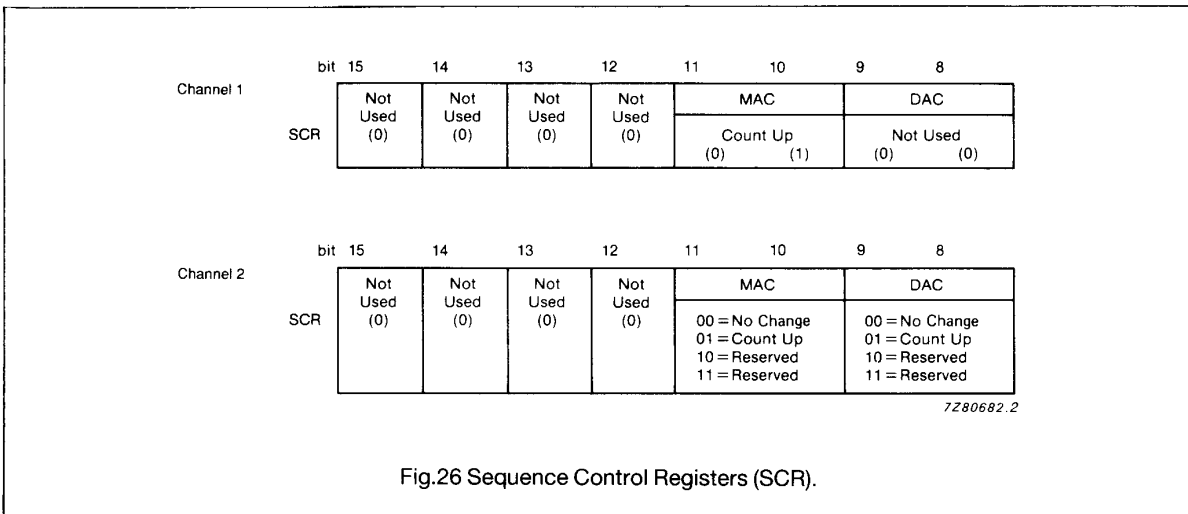
Setting this bit causes a pending start to be reset. When reading CCR this bit is always zero.

INTERRUPT ENABLE (CCR 3)

- 0 Interrupts not enabled.
- 1 Interrupts enabled.

INTERRUPT PRIORITY LEVEL (CCR 2:0)

These three bits define the priority level of the interrupt given by the channel. 000 will cause no interrupt, 001 is the lowest and 111 is the highest interrupt priority.



16/32-bit microprocessor

SCC68070

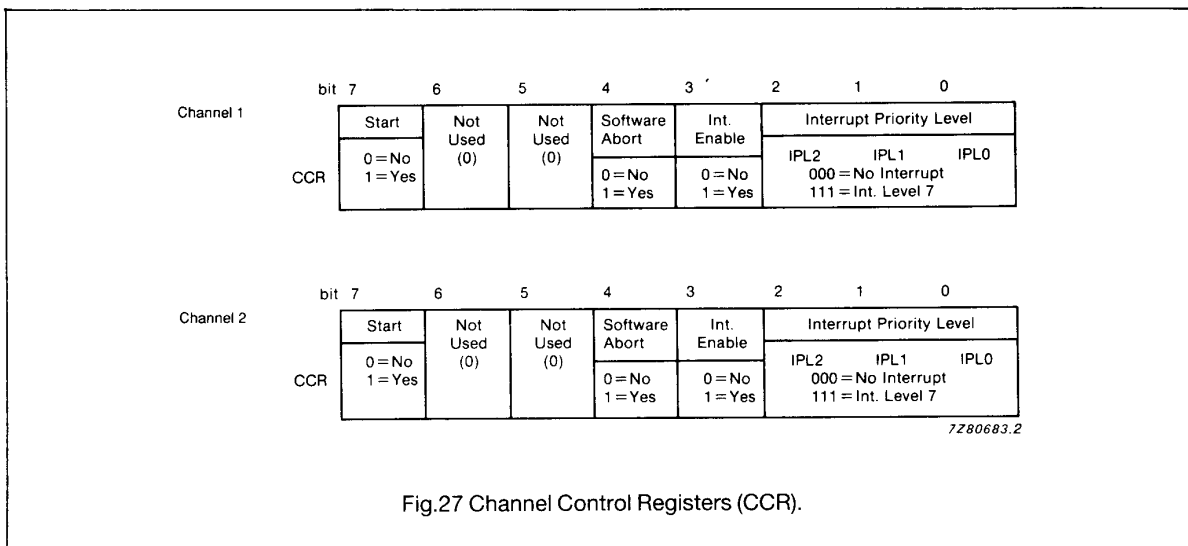


Fig.27 Channel Control Registers (CCR).

Channel Status Register (CSR)

This register is read to obtain the status of the channel.

COC BIT (CSR 15)

This bit is set after the termination (whether successful or not) of any channel operation and indicates that the DMA transfer has been completed. It must be cleared before starting another channel operation.

NDT (CSR 13)

This bit is set when the device terminates channel operation by asserting DONEN while the device was being acknowledged. This bit must be cleared before another channel operation can start.

ERR (CSR 12)

This bit is used to report that channel operation has been terminated because of an error. By reading the Channel Error Register the cause of the error can be determined. This bit must be cleared before another channel operation can start. Clearing CSR 12 also clears the Channel Error Register.

CHANNEL ACTIVE (CSR 11)

This bit is set automatically after channel operation has been started and remains set until channel operation

terminates, whereupon it is automatically reset by the channel. This bit is not effected by write operations.

Bits CSR 15, CSR 13 and CSR 12 can be cleared by writing a '1' to the appropriate bit positions of the register. Writing a '0' to these bit positions has no effect.

Channel Error Register (CER)

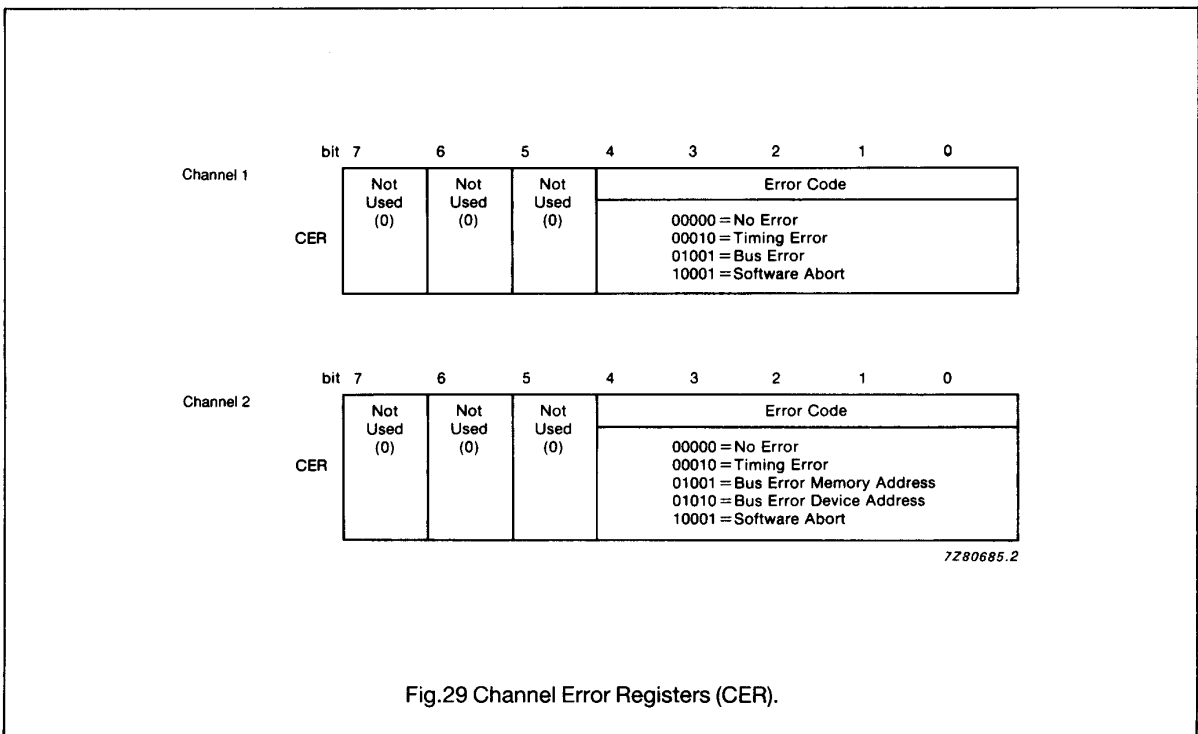
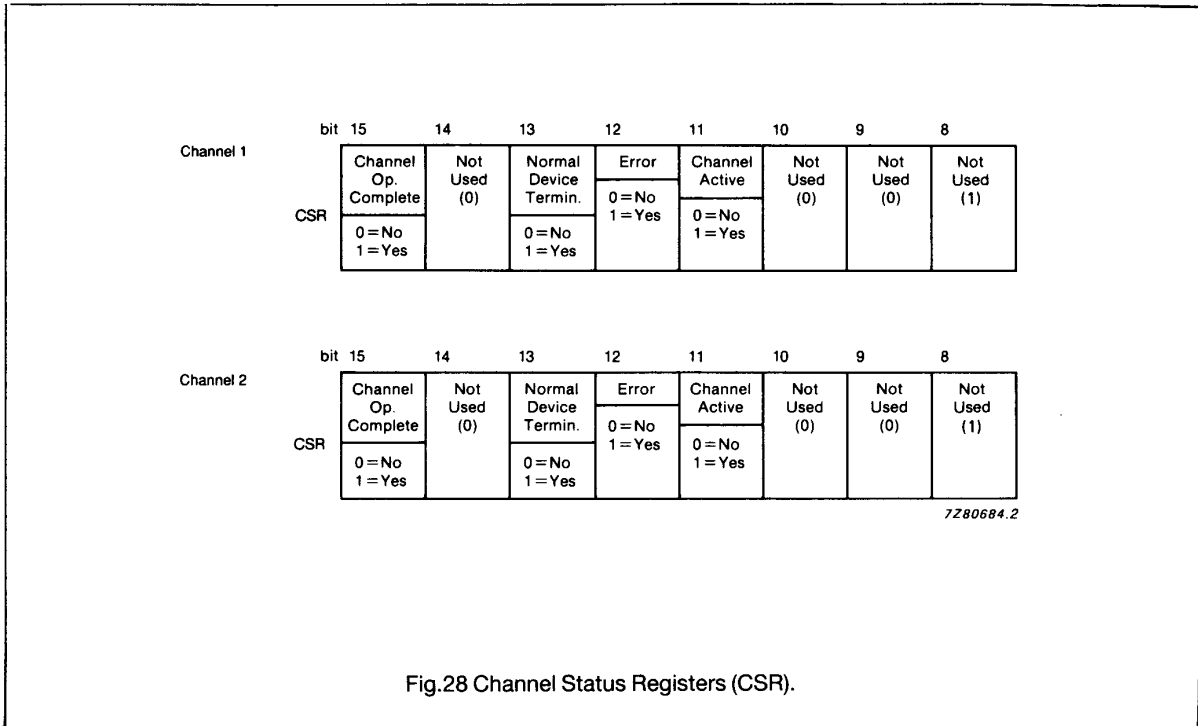
ERROR CODE (CER 4:0)

This field indicates the source of the error when the ERR bit (CSR 12) is set. Clearing CSR 12 in the Channel Status Register also clears the Channel Error Register.

- 00000 No Error.
- 00010 Timing Error. An attempt has been made to start the channel before all the bits of the Channel Status Register have been cleared.
- 01001 Bus Error memory side. A bus error (BERRN asserted without HALTN) has occurred during the cycle with MAC presenting the address.
- 01010 Bus Error device side. A bus error (BERRN asserted without HALTN) occurred during the cycle with DAC presenting the address (Channel 2 only).
- 10001 Software Abort. The channel operation was aborted by a Software Abort command (CCR 4).

16/32-bit microprocessor

SCC68070



16/32-bit microprocessor

SCC68070

Channel Priority Register (CPR)

CPR serves no function in the channel but is included only for programming compatibility with the other 68000 family DMA controllers. It contains the value '0' for Channel 1 and the value '1' for Channel 2.

Memory Address Counter (MACH, MACMH, MACML, MACL)

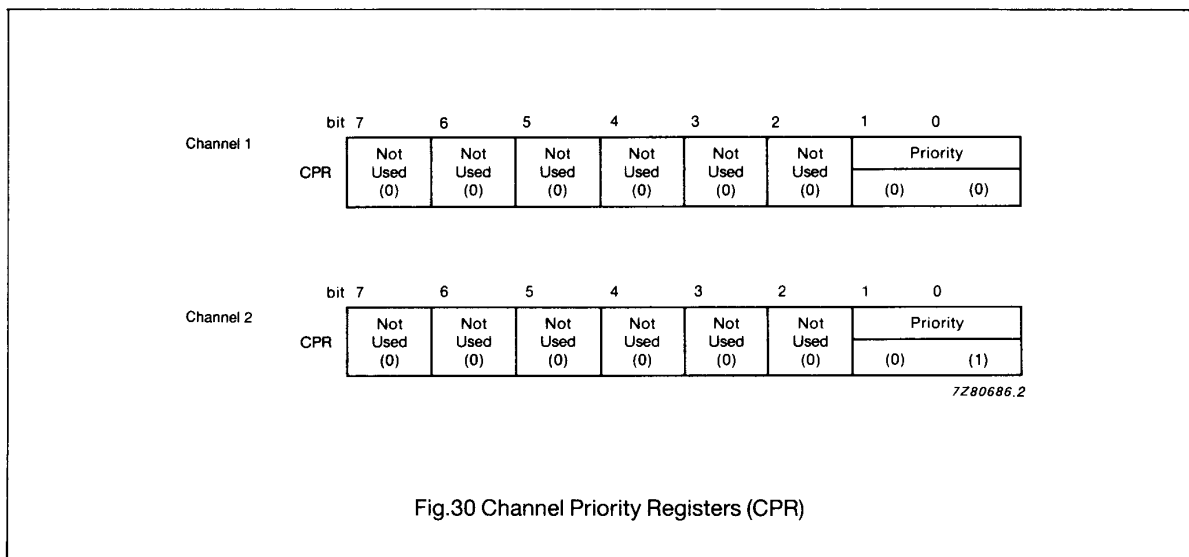
The 32-bit memory address counter defines a memory location as the source or destination of the operand to be transferred, depending on the direction of the transfer. Only the least significant 24-bits of the counter (MACMH, MACML, and MACL) are implemented in the DMA controller.

Device Address Counter (DACH, DACMH, DACML, DACL) - Channel 2 only

The 32-bit device address counter defines a device location as the source or destination of the operand to be transferred, depending on the direction of the transfer. Only the least significant 24-bits of the counter (DACMH, DACML, and DACL) are implemented in the DMA controller.

Memory Transfer Counter (MTCH, MTCL)

The 16-bit memory transfer counter defines the number of operands to be transferred by the channel.



16/32-bit microprocessor

SCC68070

Operation

The DMA controller operation has three principle phases:

- The initialization phase, during which the CPU configures the Channel Control Registers, loads the initial memory address and transfer count, and then starts the channel operation.
- The transfer phase, during which the channel accepts requests for transfers from the device, arbitrates for and acquires the bus, and provides the addressing and bus control for the transfers.
- The termination phase, which occurs after the operation has been completed when the channel reports the status of the operation.

Initialization phase

After programming the Channel Control Registers (CCR), the Memory and Device Address Counters and the Memory Transfer Counter, the CPU sets the START bit (CCR 7). The channel initializes the operation by clearing any pending requests, clearing the START bit and setting the Channel Active bit in the Channel Status Register. The channel is then ready to receive valid requests for an operation via the external REQ1N (or REQ2N) pin.

Transfer phase

Data movement between the device and memory takes place during the transfer phase in one of two ways. In single-address mode, transfers occur during a single bus cycle; in double-address mode, each transfer is performed with a read and write bus cycle.

Termination phase

The termination phase of the block transfer occurs under the conditions detailed below:

COUNT TERMINATION

During operand transfer, the channel decrements the Memory Transfer Counter (MTC). Completion of a channel transfer occurs when this counter reaches zero and the last byte or word has been moved from source to destination. The channel then notifies the device of completion by asserting the DONEN output during the last operand transfer cycle. On completion of the transfer, the Channel Active bit (CSR 11) in the Channel Status Register is cleared and the COC bit (CSR 15) is set.

DEVICE TERMINATION

The channel monitors the state of the DONEN line while acknowledging a device transfer request. If the device asserts DONEN, the channel will terminate the operation after transferring the current operand. The channel then clears bit CSR 11, and sets bits CSR 15 and CSR 13 (all these bits reside in the Channel Status Register)

SOFTWARE ABORT

The Software Abort bit (CCR 4) allows the CPU to abort the current channel operation. (See description of Channel Control Register).

Bus error treatment

If both the BERRN and HALTN signals are asserted during a DMA controller cycle, the DMA controller will enter the RERUN state. If only the BERRN signal is asserted during a DMA controller cycle, the channel stops the controller operation, releases the bus, sets the ERR (CSR 12) and COC (CSR 15) bits and clears the Channel Active bit (CSR 11); all of these reside in the Channel Status Register. The channel also sets the error code in the Channel Error Register to indicate a bus error.

Reset

Via RESETN, external sources and CPU programs can reset and initialize the DMA controller. If the DMA controller is the bus master when reset is detected, it releases the bus and resets all the bits of the Channel Status Register (except CSR 8) to zero.

Interrupts

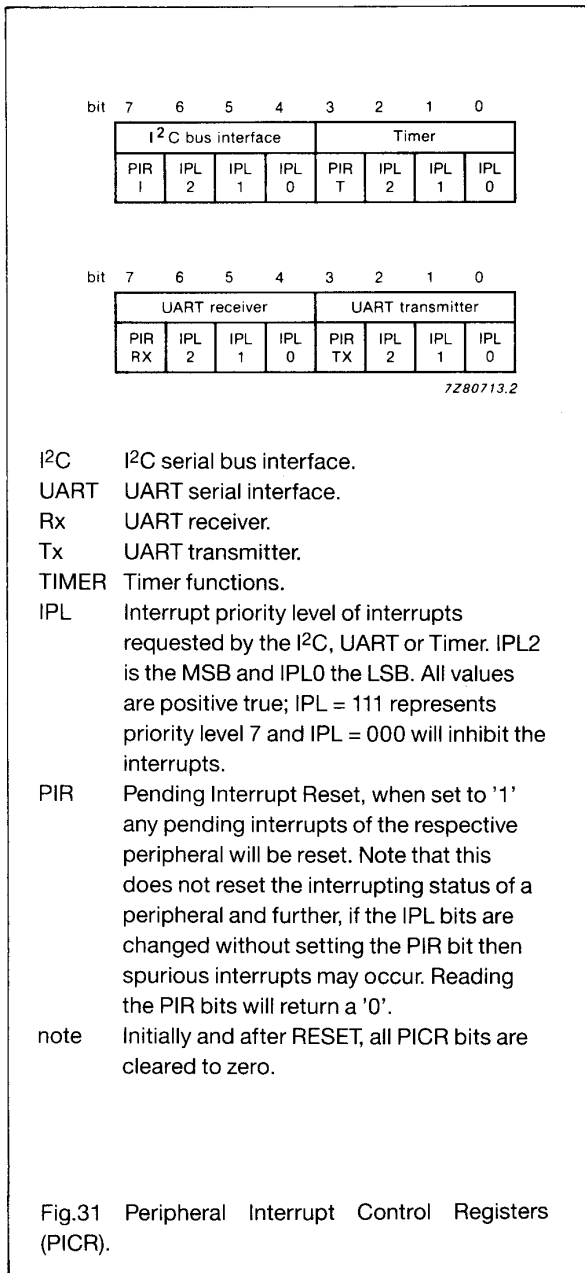
If the Interrupt Enable bit (CCR 3) is set, the channel will send an interrupt when it terminates an operation (COC bit set - CSR 15). The priority level of the interrupt is given by the IPL bits in the Channel Control Register. During the interrupt acknowledge cycle, the channel requests an autovector and thus the IPL bits correspond directly to the vector used.

16/32-bit microprocessor

SCC68070

PERIPHERAL INTERRUPT CONTROL

The I²C and UART serial interfaces and the Timer, use a common set of Peripheral Interrupt Control Registers (PICR). These registers are memory mapped on the on-chip bus and communicate with the CPU of SCC68070.



THE I²C SERIAL BUS INTERFACE

The SCC68070 has a serial I/O interface so that it can communicate with other devices via the I²C-bus. The I²C-bus can be used in master or slave mode, and can be connected to a maximum of 128 different peripheral ICs, each with a unique device address. Maximum transmission speed is 100 kbits/s.

Communication with the bus is via two dedicated pins, SCL the serial clock pin and SDA the serial data pin. The interface can generate interrupts with priorities programmed to one of seven levels. A complete data transfer is shown in Fig.32.

Operating modes

The CPU can operate in the following modes with the serial I²C-bus.

- master transmitter (MTX)
- master receiver (MRX)
- slave transmitter (STX)
- slave receiver (SRX)

The I²C-bus I/O registers

The communication between the CPU and the I²C-bus interface is via a set of registers and an interrupt request facility. All I²C registers are accessible by read or write operations. The data and information controlling the operation of the interface is stored in the following registers (these are fully transparent and memory mapped to the CPU):

Table 21 Register memory map.

Base address 8000 2000 (HEX)

| A3 | A2 | A1 | A0 | I ² C REGISTER |
|----|----|----|----|-------------------------------|
| 0 | 0 | 0 | 0 | reserved |
| 0 | 0 | 0 | 1 | Data Register (IDR) |
| 0 | 0 | 1 | 0 | reserved |
| 0 | 0 | 1 | 1 | Address Register (IAR) |
| 0 | 1 | 0 | 0 | reserved |
| 0 | 1 | 0 | 1 | Status Register (ISR) |
| 0 | 1 | 1 | 0 | reserved |
| 0 | 1 | 1 | 1 | Control Register (ICR) |
| 1 | 0 | 0 | 0 | reserved |
| 1 | 0 | 0 | 1 | Clock Control Register (ICCR) |

16/32-bit microprocessor

SCC68070

Data Register (IDR)

The data register IDR performs the conversion between the serial and parallel data formats. Data to be transmitted is loaded into IDR by the CPU and then shifted out serially (MSB first), and data received on the serial bus is shifted into IDR (MSB first)

Address Register (IAR)

The address register holds the slave address allocated to the device in its 7 MSB's. It is only written to by the CPU and remains unchanged until rewritten. The LSB is the Always Selected bit (ALS) and when set to '1' will disable the address recognition and the interpretation of the R/W bit of the first byte. Thus, the I²C-bus interface will transfer in the Free Data Format and will respond to each data transfer.

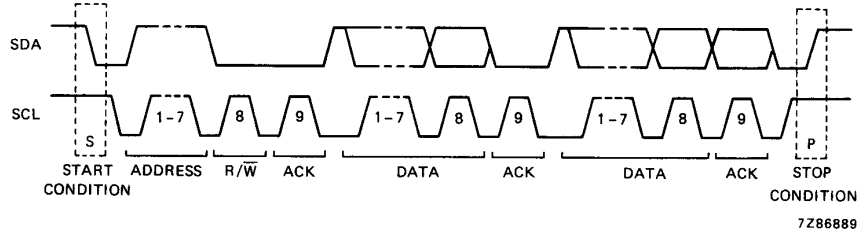


Fig.32 A complete data transfer on the I²C-bus.

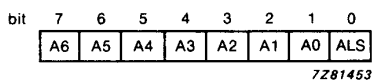


Fig.33 I²C Address Register (IAR).

16/32-bit microprocessor**SCC68070****Status Register (ISR)**

The Status Register contains all the information concerning the status of the I²C-bus interface. All bits can be written to or read, by the CPU. The functions of the status bits, illustrated in Fig.34, follow:

MST = MASTER

If MST is '1', the I²C-bus interface is in the master mode and it generates clock pulses on SCL for transmission/reception timing of serial data.

If MST is '0', the I²C-bus is in the slave mode and clock pulses are received from the master on SCL.

TRX = TRANSMITTER

If TRX is '1', the I²C-bus interface is in the transmitter mode and data in the IDR is shifted out onto the data line SDA, synchronized with the clock pulses on SCL.

If TRX is '0', the I²C-bus interface is in the receiver mode and data on the data line SDA is shifted into the IDR synchronized with the clock pulses on SCL.

BB = BUSY

This bit indicates the state of the serial bus.

If BB is '0', the bus is free.

If BB is '1', the bus is busy.

PIN = PENDING INTERRUPT NOT

The PIN bit is set to '0' every time an I²C-bus interrupt is requested.

Any access to IDR will set the PIN bit to '1'.

AL = ARBITRATION LOST

The AL bit generally indicates the detection of an error. It is set to '1' when:

- a data error occurs in the transmitter mode
- the CPU tries to write to the ISR from the slave mode when it is not selected and the bus-busy flag is already set to '1'

The AL bit is reset to '0' by any access to IDR.

AAS = ADDRESSED AS SLAVE

AAS is set to '1' when the address comparator recognizes either its own slave address, the general call address (8 zeros) or the first byte in the free data format that has been received.

AAS is reset to zero by any access to IDR.

AD0 = ADDRESS ZERO

AD0 is set to '1' if the address comparator detects the general call address (8 zeros). It is reset to '0' when either a START or STOP condition is detected.

LRB = LAST RECEIVED BIT

When the interface is in the transmitter mode the LRB bit contains the receiver acknowledge bit. LRB is '0' when the reception of the transmission has been acknowledged.

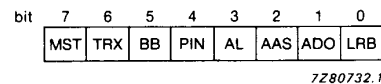


Fig.34 I²C Status Register (ISR).

Control Register (ICR)

Some additional functions of the interface are provided by the Control Register (ICR) as shown in Fig.35.

SEL = SELECTED

The SEL bit is a flag set (together with the AAS bit in the Status Register) by the interface logic when in the slave mode, and remains set during the whole transfer. It is reset when a STOP, START or repeated Start condition is detected.

ESO = ENABLE I²C-BUS

When the ESO bit is set the I²C-bus is enabled and when reset the I²C-bus is disabled. Only the CPU can alter the ESO bit.

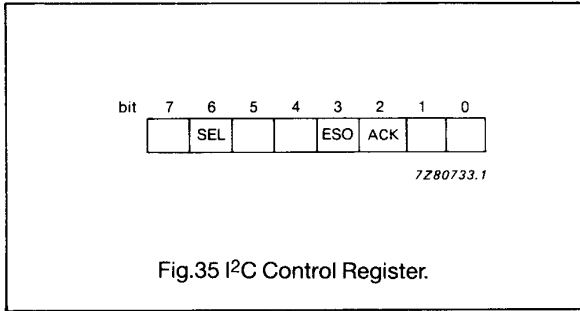
16/32-bit microprocessor

SCC68070

ACK = ACKNOWLEDGE

This bit determines the polarity of the acknowledge that a receiver sends after correct reception of a byte.

If ACK = '1', reception will be acknowledged by a '0' bit.
If ACK = '0', reception will not be acknowledged, a '1' bit is sent.



Clock Control Register (ICCR)

By programming the 5 LSBs of the Clock Control Register, the frequency of SCL and SDA can be adapted to the needs of the I²C-bus or the SCC68070's system clock. After initialization or RESET, bits CK4 to CK0 are cleared to zero and must be programmed to a non-zero value before the I²C-bus is enabled.

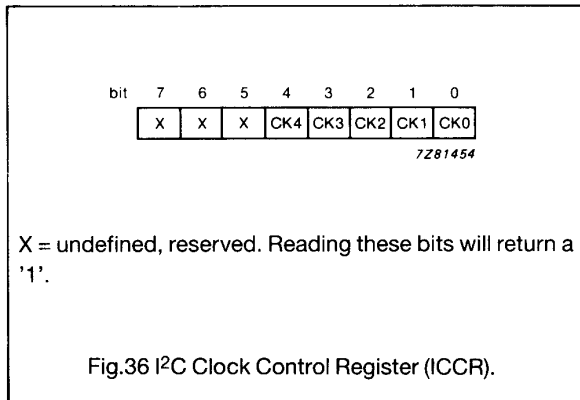


Table 22 I²C-bus interface divisors.

| CK4-CK0 (HEX) | DIVISOR | ≈ SCL FREQ. (KHZ) (1) | ≈ SCL FREQ. (KHZ) (2) |
|---------------|---------|------------------------|------------------------|
| 0 | illegal | - | - |
| 1 | 78 | 126.025 ⁽³⁾ | 192.308 ⁽³⁾ |
| 2 | 90 | 109.222 ⁽³⁾ | 166.667 ⁽³⁾ |
| 3 | 102 | 96.372 | 147.059 ⁽³⁾ |
| 4 | 126 | 78.015 | 119.048 ⁽³⁾ |
| 5 | 150 | 65.533 | 100.000 |
| 6 | 174 | 56.494 | 86.207 |
| 7 | 198 | 49.464 | 75.758 |
| 8 | 246 | 39.959 | 60.976 |
| 9 | 294 | 33.435 | 51.020 |
| A | 342 | 28.742 | 43.860 |
| B | 390 | 25.205 | 38.462 |
| C | 486 | 20.266 | 30.864 |
| D | 582 | 16.890 | 25.773 |
| E | 678 | 14.498 | 22.124 |
| F | 774 | 12.700 | 19.380 |
| 10 | 996 | 10.175 | 15.060 |
| 11 | 1158 | 8.488 | 12.953 |
| 12 | 1350 | 7.281 | 11.111 |
| 13 | 1542 | 6.374 | 9.728 |
| 14 | 1926 | 5.103 | 7.788 |
| 15 | 2310 | 4.255 | 6.494 |
| 16 | 2694 | 3.648 | 5.568 |
| 17 | 3078 | 3.193 | 4.873 |
| 18 | 3846 | 2.555 | 3.900 |
| 19 | 4614 | 2.130 | 3.251 |
| 1A | 5382 | 1.826 | 2.787 |
| 1B | 6150 | 1.598 | 2.439 |
| 1C | 7686 | 1.278 | 1.952 |
| 1D | 9222 | 1.065 | 1.627 |
| 1E | 10758 | 0.913 | 1.394 |
| 1F | 12294 | 0.799 | 1.220 |

Notes to Table 22

1. The crystal or clock input frequency = 19.6608 MHz.
2. The crystal or clock input frequency = 30 MHz.
3. The maximum bus clock frequency in an I²C system is specified as 100 kHz.

16/32-bit microprocessor**SCC68070****UART SERIAL INTERFACE**

A UART (Universal Asynchronous Receiver/Transmitter) interface is included on-chip and functions like a subset of the UART's 2642, 2661 and 2691.

The UART interfaces directly with the CPU and can be used in either polled or interrupt driven modes. It accepts programmed instructions from the CPU whilst supporting asynchronous serial data communication in either full or half-duplex mode. The interface then converts data received from the CPU into a serial form for transmission, and simultaneously it can receive serial data and convert it into parallel data as input to the CPU. Two bit rate generators can be programmed to generate transmit/receive bit rates by either using the SCC68070's system clock, or accepting an external clock via the XCKI input.

Table 23 UART register addressing

Base address 8000 2011 (HEX)

| A3 | A2 | A1 | A0* | REGISTER |
|----|----|----|-----|---------------------------------|
| 0 | 0 | 0 | 1 | Mode Register (UMR) |
| 0 | 0 | 1 | 1 | Status Register (UMR) |
| 0 | 1 | 0 | 1 | Clock Select Register(UCSR) |
| 0 | 1 | 1 | 1 | Command Register (UCR) |
| 1 | 0 | 0 | 1 | Transmit Holding Register (UTH) |
| 1 | 0 | 1 | 1 | Receive Holding Register (URH) |
| 1 | 1 | 0 | 1 | undefined, reserved |
| 1 | 1 | 1 | 1 | undefined, reserved |

* All allocations with A0 = 0 are undefined, reserved.

Programming

Before initiating data communication, the UART operation mode must be programmed by writing to the mode, Clock Select and Command Registers. The UART interface can be re-configured at any time during program execution but when writing to the Mode and Clock Select Registers the transmitter and receiver should be disabled. The Mode Register defines the general operational characteristics of the interface, while the Command Register controls the operation within the basic framework. Using the Clock Select Register, the bit rates for transmitter and receiver can be set and the result of the operation displayed in the Status Register. Certain bits of these registers are cleared when either a RESET input is applied, when a RESET instruction is performed by the CPU, or when special reset commands are programmed into the UART interface.

16/32-bit microprocessor

SCC68070

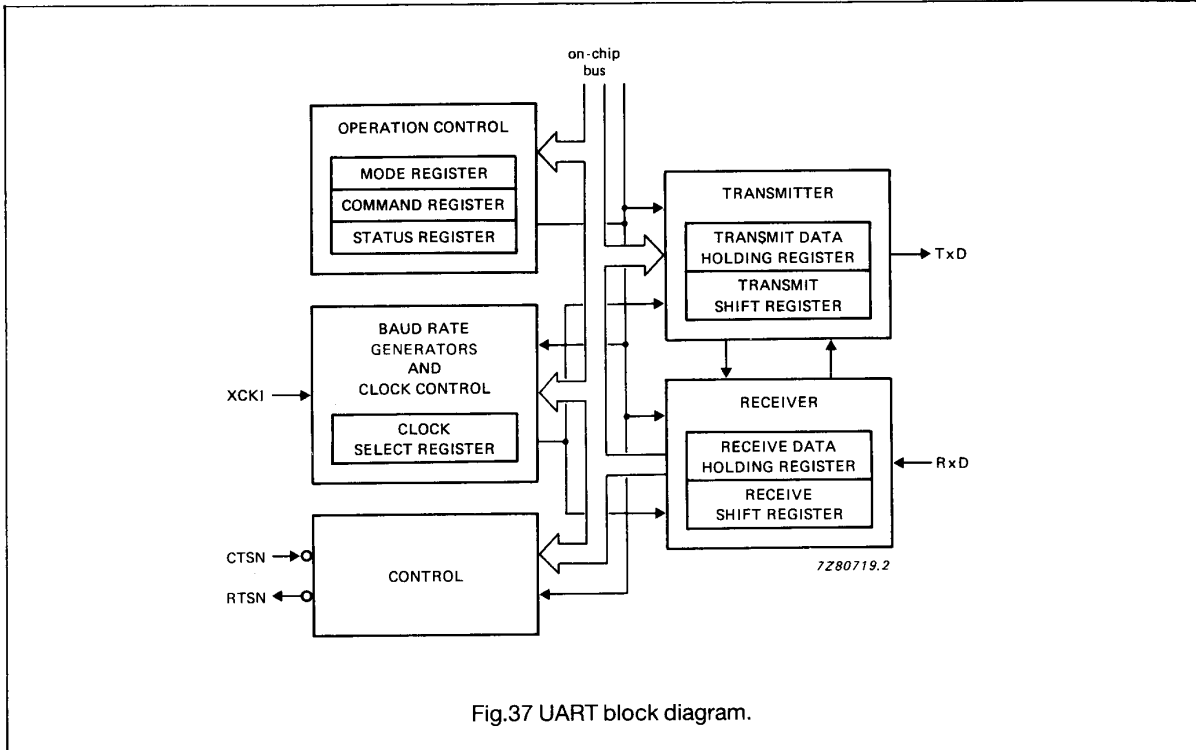


Fig.37 UART block diagram.

Mode Register (UMR)

| | |
|---|--|
| <p>UMR 0 selects the character length (7 or 8 bits). This does not include the parity bit (if programmed) or the START/STOP bits.</p> <p>UMR 1 selects the number of STOP bits, either 1 or 2.</p> <p>UMR 2 selects either odd or even parity when parity has been enabled by UMR 3.</p> <p>UMR 3 controls the parity generation and when enabled a parity bit is added to the transmitted character and the receiver performs a parity check on the incoming data.</p> <p>UMR 4 determines if the CTSN input controls the operation of the transmitter. If set to '0', CTSN has no influence on the transmitter. If set to '1', the transmitter monitors the state of CTSN every time it</p> | <p>is ready to send a character and delays transmission until CTSN has been asserted</p> <p>UMR 5 not used.</p> <p>UMR 7:6 these bits determine the operation mode of the UART interface.</p> <p>UMR 7:6 = 00 is the normal mode, with the transmitter and receiver operating independently.</p> <p>UMR 7:6 = 01 places the channel in the auto echo mode which automatically retransmits the received data.</p> <p>UMR 7:6 = 10 selects local loopback mode.</p> <p>UMR 7:6 = 11 selects the remote loopback mode.</p> <p>All bits of this register are cleared by a RESETN signal or a RESET instruction issued by the CPU, with the exception of bit 5 which is not used but returns a '1' when read.</p> |
|---|--|

16/32-bit microprocessor

SCC68070

| | | | | | | | |
|--|----------|---|------------------------------|---------------------|---------------------------------------|----------------------------------|---|
| bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Channel mode | Not used | CTS enable TxD | Parity control | Parity type | Stop bit length | Character length | |
| 00 = normal 01 = auto echo 10 = local loopback 11 = remote loopback | (1) | 0 = no CSTN control 1 = CSTN control TxD | 0 = inhibited 1 = enabled | 0 = odd 1 = even | 0 = one stop bit 1 = two stop bits | 0 = seven bits 1 = eight bits | |

7Z81455

Fig.38 UART Mode Register (UMR).

Clock Select Register (UCSR)

The UCSR allows selection of the clock source and baud rate for receiver and transmitter; as shown in Fig.39. The baud rates given are generated when either a 19.6608 MHz clock is used as the SCC68070's system clock (source = internal) or a 4.9152 MHz clock is applied to XCKI (source = external). Other frequencies will give a different set of baud rates. Note that when using the SCC68070's internal clock, it is pre-divided by 4.

UCSR 7 selects the clock source for the receiver and transmitter baud rates. After RESET, the clock source is the on-chip clock and with a 19.6608 MHz crystal, the listed baud rates are possible.

An external clock source (XCKI) is selected if UCSR 7 is '1'. The maximum frequency that can be applied to XCKI is 10 MHz.

All bits of this register are cleared by a RESETN signal or by a RESET instruction issued by the CPU, with the exception of Bit 3 which is not used but returns a '1' when read.

| | | | | | | | |
|---|-----------------------|---------------|----------|--------------------------|---------------|---|---|
| bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Clock source | Receiver clock select | | Not used | Transmitter clock select | | | |
| | bit rate | divisor | | bit rate | divisor | | |
| 0 = internal (default after RESETN) | 000 = | 75 baud 65536 | (1) | 000 = | 75 baud 65536 | | |
| | 001 = | 150 32768 | | 001 = | 150 32768 | | |
| | 010 = | 300 16384 | | 010 = | 300 16384 | | |
| | 011 = | 1200 4096 | | 011 = | 1200 4096 | | |
| | 100 = | 2400 2048 | | 100 = | 2400 2048 | | |
| | 101 = | 4800 1024 | | 101 = | 4800 1024 | | |
| 1 = external | 110 = | 9600 512 | 110 = | 9600 512 | | | |
| | 111 = | 19200 256 | 111 = | 19200 256 | | | |

7Z81456

Fig.39 UART Clock Select Register (UCSR).

16/32-bit microprocessor

SCC68070

Command Register (UCR)

The UCR is used to write commands to the UART.

UCR 6:4 = Miscellaneous commands. The encoded value of this field may be used to specify a single command as follows.

- 000 No command
- 001 No command
- 010 Reset receiver. Resets the receiver as if a hardware reset has been applied.
- 011 Reset transmitter. Resets the transmitter as if a hardware reset has been applied.
- 100 Reset error status. Clears the received break, parity error, framing error and overrun error bits in the Status Register USR 7:4.
- 101 No command.
- 110 Start break. Forces the TxD output LOW (spacing).
- 111 Stop break. The TxD line will go HIGH (marking) within two bit times. TxD will remain HIGH for

one bit time before the next character, if any, is transmitted.

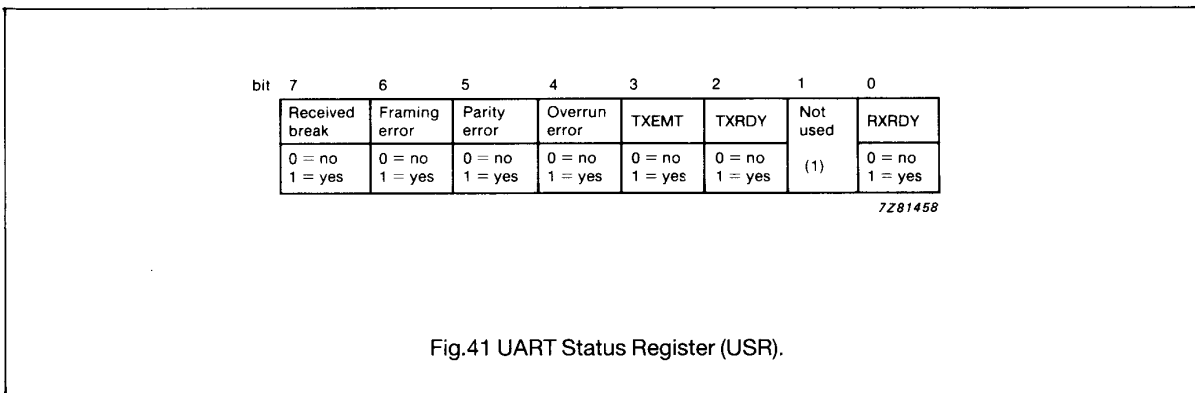
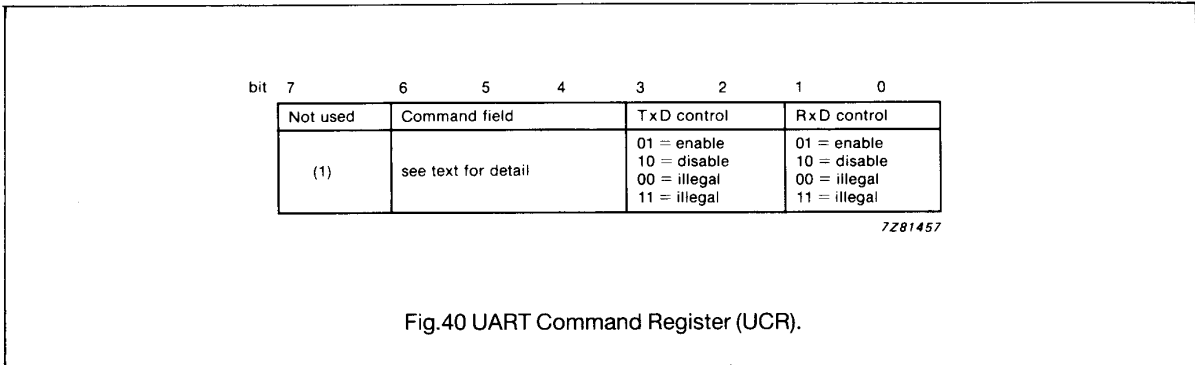
All bits of this register are cleared by a RESETN signal or by a RESET instruction issued by the CPU, with the exception of Bit 7 which is not used but returns a 1 when read.

Status Register (USR).

The Status Register can be read by the CPU to determine the condition of an enabled receiver or transmitter.

All bits of this register are cleared by a RESETN or by a RESET instruction issued by the CPU, except bit 1 which is not used and returns a '1' when read.

- TxE_MT = Transmitter Empty
- TxR_DY = Transmitter Ready
- RxR_DY = Receiver Ready



16/32-bit microprocessor

SCC68070

TIMER

The Timer comprises a 16-bit reference timer with an auto-reload register and two identical (independently function-programmable) 16-bit registers. The clock period of the reference timer and hence the maximum resolution is $96/CLKOUT(MHz) \mu s$. Two programmable I/O lines provide the necessary connection to external circuitry. Three modes can be selected:

- match or pulse output mode which changes the output state when there is a match between the reference and register values
- count mode which counts external events that occur at the T1 (T2) input
- capture mode which stores the reference timer value in a capture register when an external event occurs at the T1 (T2) input

Any transition on the inputs to T1 or T2 can be programmed as an external event.

Timer programming

The CPU can read from or write to all the timer registers and they can also be accessed "on the fly". The address map of the timer registers is shown in Table 24.

Table 24 Timer registers address map.

Base address 8000 2020 (HEX)

| A3 | A2 | A1 | A0 | REGISTER |
|----|----|----|----|------------------------------|
| 0 | 0 | 0 | 0 | Timer Status Register (TSR) |
| 0 | 0 | 0 | 1 | Timer Control Register (TCR) |
| 0 | 0 | 1 | 0 | Reload Register High (RRH) |
| 0 | 0 | 1 | 1 | Reload Register Low (RRL) |
| 0 | 1 | 0 | 0 | Timer 0 High (T0H) |
| 0 | 1 | 0 | 1 | Timer 0 Low (T0L) |
| 0 | 1 | 1 | 0 | Timer 1 High (T1H) |
| 0 | 1 | 1 | 1 | Timer 1 Low (T1L) |
| 1 | 0 | 0 | 0 | Timer 2 High (T2H) |
| 1 | 0 | 0 | 1 | Timer 2 Low (T2L) |

Timer Control Register (TCR)

- EVENT** Control for external events monitored to trigger a function in Timer registers T1 or T2.
- 00 Input inhibited.
 - 01 LOW-to-HIGH transitions will be monitored.
 - 10 HIGH-to-LOW transitions will be monitored.
 - 11 Any transition will be monitored.
- MODE** Control for the function of Timer registers T1 or T2.
- 00 Timer inhibited.
 - 01 Match Mode. A match between the reference timer T0 with the respective Timer register will reset output T1 or T2. Each overflow of T0 will set output T1 or T2. The I/O port of the Timer is automatically switched to output mode.
 - 10 Capture Mode. When an external event occurs (as described above), the contents of the reference timer T0 will be stored in the Timer register T1 or T2.
 - 11 Event Counter Mode. When an external event occurs the timer register T1 or T2 will be incremented.

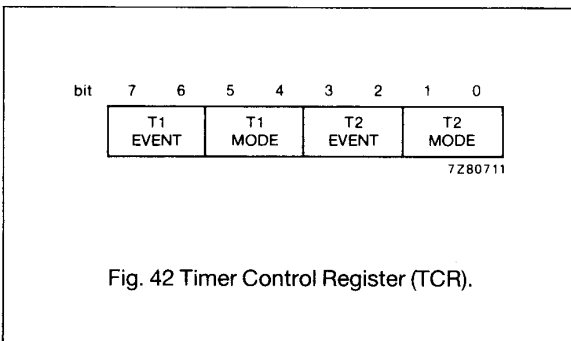


Fig. 42 Timer Control Register (TCR).

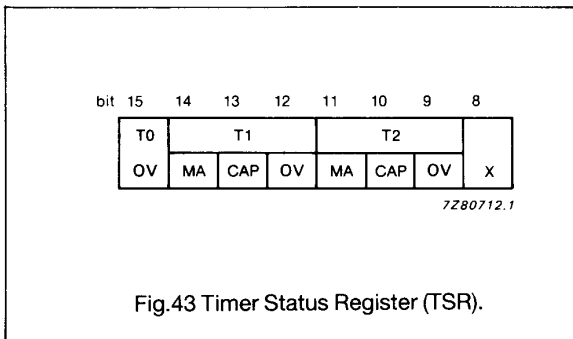
16/32-bit microprocessor**SCC68070****Timer Status Register (TSR)**

The Timer Status Register indicates which timer and what specific event occurred that caused an interrupt (if enabled) and can be read by the CPU. After being read, each bit of this register should be reset by software as an acknowledge of the read because the status bits are automatically set but are not reset by the Timer. To reset each bit a '1' must be written to the appropriate bit position of the register.

- OV** Overflow. The Timer counts from FFFF(HEX) to 0000(HEX). This bit will be reset by timers T1 and T2 in event-counter mode only.
- MA** Match. A match between the value stored in Timer registers T1 or T2 and the value of the continuous timer T0 has occurred (in match mode).
- CAP** Capture. When an external event occurs the current value of the continuous timer T0 is stored in Timer T1 or T2 (in capture mode).
- X** Undefined, reserved.

Reference timer

The reference timer T0 will increment by 1 (starting from the value initially loaded into T0H and T0L). Using a crystal frequency of 19.6608 MHz the SCC68070 will increment every 9.766 μ s (or 96 CKOUT cycles). When T0 reaches FFFF, the OV flag in the status register is set, and the reload register (RR) is loaded into T0. T0 will then start incrementing again until the next overflow occurs. A 30 MHz XTAL frequency will increment the timer every 6.4 μ s, independently of the XCLI frequency. A 35 MHz XTAL frequency will increment the timer every 5.49 μ s.



16/32-bit microprocessor**SCC68070****LIMITING VALUES**

In accordance with the Absolute Maximum System (IEC 134).

| SYMBOL | PARAMETER | MIN. | MAX. | UNIT |
|------------|--|----------|--------------|--|
| V_I | Input voltage on any pin with respect to ground (V_{SS}) | | -0.3 | |
| I_I, I_O | Input, output current | - | ± 10 | mA |
| P_{tot} | Total power dissipation | - | 2 | W |
| T_{stg} | Storage temperature range | -55 | + 150 | $^{\circ}\text{C}$ |
| T_{amb} | Operating ambient temperature ranges* | 0 -40 | + 70 + 85 | $^{\circ}\text{C}$ $^{\circ}\text{C}$ |

* Devices are available in two temperature ranges; see Ordering information.

Notes to Limiting Values and Electrical characteristics

- Stresses above those listed in the Absolute Maximum System may cause permanent damage to the device. These are stress ratings only and do not mean that the device will operate at these or other conditions above those given in the operation section.
- For operating at elevated temperatures, the device must be derated based on a 150 $^{\circ}\text{C}$ maximum junction temperature.
- This product contains circuitry specifically designed to protect its internal devices from excessive static charge. Nevertheless it is recommended that conventional precautions be taken to avoid applying any voltage above the rated maxima.
- Parameters are valid over specified temperature range.
- All voltages are measured with ground as reference (GRD). For testing, all input signals swing between 0.4 and 2.4 V with a transmission time of 5 ns maximum. All time measurements are made with input and output voltages of 0.8 and 2.0 V as appropriate.
- On clock input XTAL1 when an external clock is used.
- All timing measurements have CKOUT as a reference for both internal oscillator and external clock input modes. The device has been designed to be used with a 35 MHz crystal but the minimum crystal frequency specified is 8 MHz. All timing measurements except number 1 are specified at 19.6608, 25, 30 and 35 MHz.
- Actual value depends on clock period.
- After V_{DD} has been applied for 100 ms.
- If the asynchronous setup time (#41A) requirements are met, the DTACKN LOW-to-data setup time (#31) requirements can be ignored. The data must only satisfy the data-in to clock-LOW setup time (#27) for the following cycle.
- If the asynchronous setup time (#41A) requirements are met, for both DTACKN and BERRN, then #42 may be 0 ns.
- All timing diagrams should only be referred to in regard to edge-to-edge measurements of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to the functional description and related diagrams for device operation.

16/32-bit microprocessor**SCC68070****DC CHARACTERISTICS**

$V_{DD} = 5.0 \text{ V} \pm 10\%$; $V_{SS} = 0 \text{ V}$; $T_{amb} = 0 \text{ to } +70 \text{ }^\circ\text{C}$ or $-40 \text{ to } +85 \text{ }^\circ\text{C}$, dependent on type number. (See Figs.44 to 47 and notes 4 and 5).

| SYMBOL | PARAMETER | CONDITIONS | MIN. | MAX. | UNIT |
|-----------|--|---|--------------|----------|---------------|
| V_{IH} | Input voltage HIGH, all inputs except XTAL1, XTAL2, SDA, SCL | | 2.0 | V_{DD} | V |
| V_{IH1} | XTAL1 | $V_{DD} = 5.5 \text{ V}$ | $0.8V_{DD}$ | V_{DD} | V |
| V_{IH2} | SDA, SCL | - | 3.0* | V_{DD} | V |
| V_{IL} | Input voltage LOW, all inputs except SDA, SCL | - | $V_{SS}-0.3$ | 0.8 | V |
| V_{IL2} | SDA, SCL | $V_{DD} = 4.5 \text{ V}$ | $V_{SS}-0.3$ | 1.5 | V |
| I_{LI} | Input leakage current RXD, CTSN, XCKI, DTACKN, INT1N, INT2N, REQ1N, REQ2N, RDYN, IN2N, IN4N, IN5N, NMIN, AVN, XTAL1 | $V_{DD} = 5.25 \text{ V}$ $V_{IN} = 5.25 \text{ V}^{**}$ | - | 20 | μA |
| I_{TSI} | 3-state (off-state) input current A1-A23, D0-D15, ASN, LDSN, R/WN, UDSN, T1, T2 | $V_{IN} = 2.4/0.4 \text{ V}$ | - | 20 | μA |
| I_{ODI} | Open-drain (off-state) input current BGACKN, RESETN, HALTN, BERRN, DTCN, DONEN, SCL, SDA | $V_{DD} = 5.25 \text{ V}$ | - | 20 | μA |
| V_{OH} | Output voltage HIGH A1-A23, D0-D15, ASN, BGN, LDSN, R/ WN, UDSN, T1, T2, TXD, RTSN, ACKN1N, ACKN2N, IACKN2,4,5,7N | $I_{OH} = 400 \mu\text{A}$ | 2.4 | - | V |
| V_{OH} | CKOUT | $I_{OH} = 400 \mu\text{A}$ | $0.8V_{DD}$ | - | V |
| V_{OL} | Output voltage LOW HALTN, BERRN, IACKN2,4,5,7, A1-A23, BGN, BGACKN, ACKN1N, ACKN2N, RESETN, T1, T2, RTSN, ASN, D0-D15, LDSN, R/WN, UDSN, DTCN, DONEN | $I_{OL} = 3.2 \text{ mA}$ | - | 0.5 | V |
| V_{OL} | CKOUT | $I_{OL} = 3.2 \text{ mA}$ | - | 0.45 | V |
| V_{OL} | SDA, SCL | $I_{OL} = 3.0 \text{ mA}$ | - | 0.45 | V |
| I_{DD} | Current consumption | CKOUT = 10 MHz | - | 75 | mA |
| | | CKOUT = 15 MHz | - | 112 | mA |
| | | CKOUT = 17.5 MHz | - | 130 | mA |
| C_i | Input capacitance | $V_{IN} = 0 \text{ V}$, $T_{amb} = 25 \text{ }^\circ\text{C}$ frequency = 1 MHz | - | 20 | pF |

* Not tested, applied by external pullups.

** V_{IN} = enforced voltage.

16/32-bit microprocessor

SCC68070

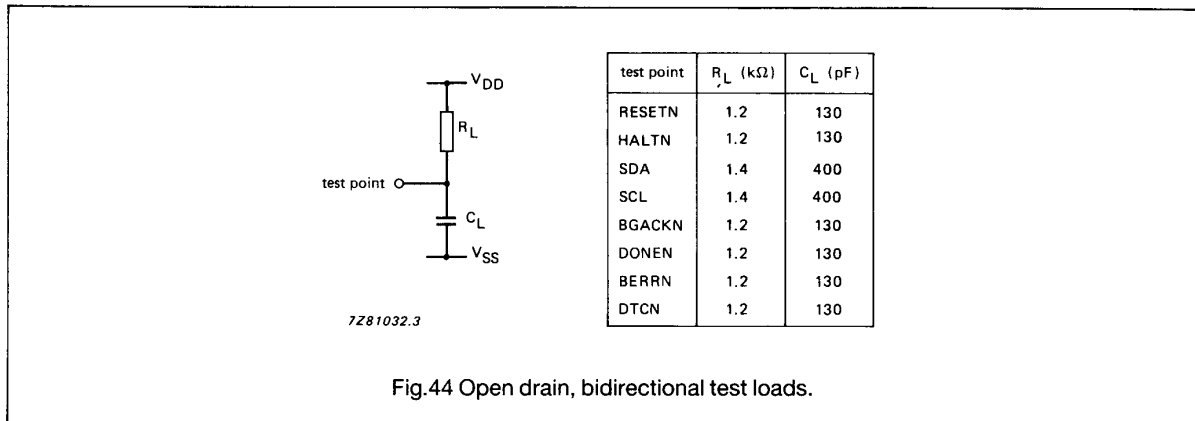


Fig.44 Open drain, bidirectional test loads.

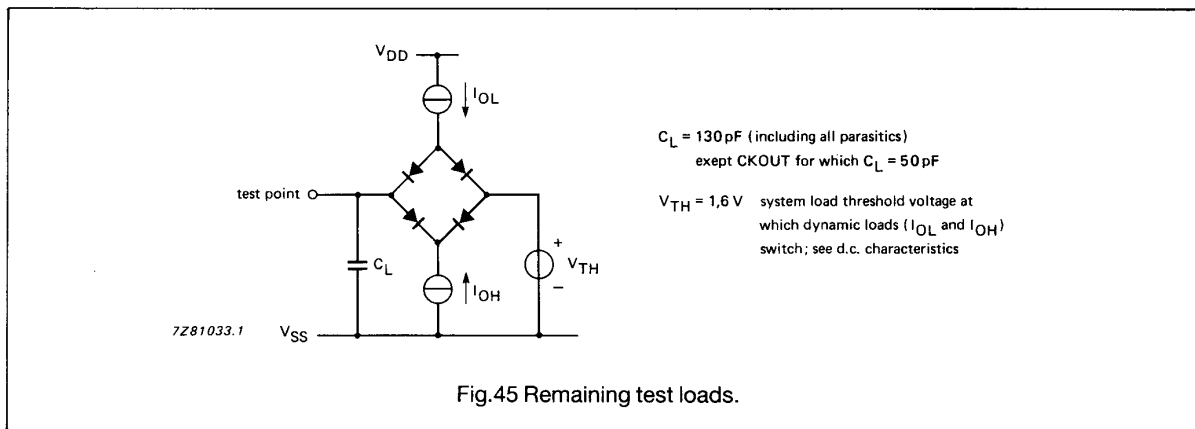
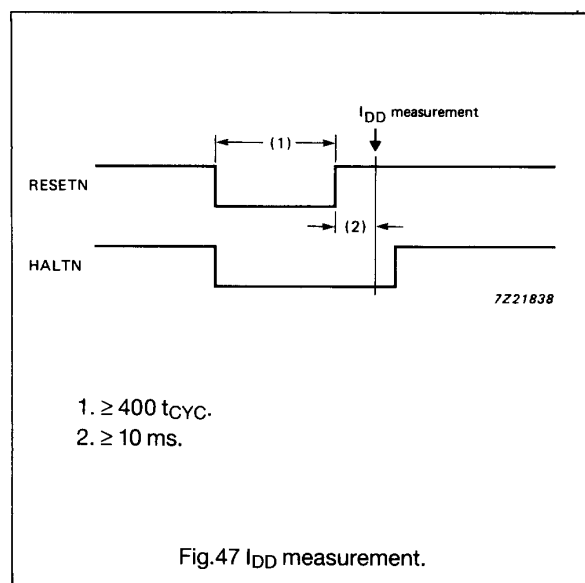
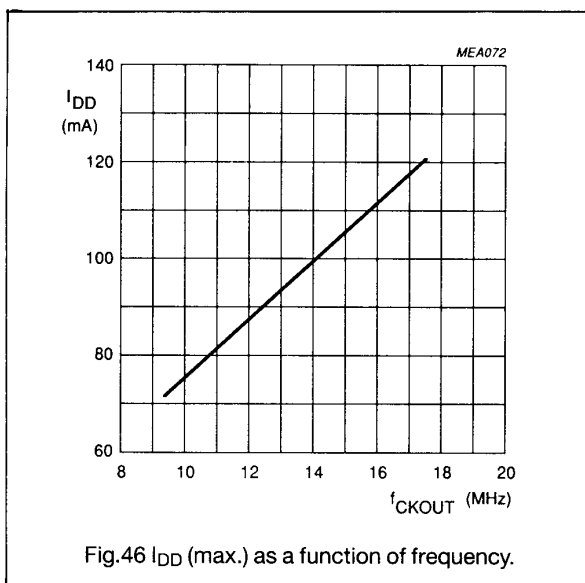


Fig.45 Remaining test loads.



16/32-bit microprocessor

SCC68070

AC CHARACTERISTICS

$V_{DD} = 5\text{ V} \pm 10\%$, $V_{SS} = 0\text{ V}$, $T_{amb} = 0\text{ to } +70\text{ }^{\circ}\text{C}$ or $-40\text{ to } +85\text{ }^{\circ}\text{C}$ (dependent on type number), C_{load} on CKOUT = 50 pF (see Figures 48 to 51).

| SYMBOL | PARAMETER | NO. | $f_{XTAL1}=19.6\text{ MHz}$ | | $f_{XTAL1}=25\text{ MHz}$ | | $f_{XTAL1}=30\text{ MHz}$ | | $f_{XTAL1}=35\text{ MHz}$ | | UNIT |
|-------------|---|------------------|-----------------------------|------|---------------------------|------|---------------------------|------|---------------------------|------|------|
| | | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | |
| t_{CYC} | Crystal or input clock period | 1 | 50 | 125 | 40 | 125 | 33 | 125 | 28.5 | 125 | ns |
| t_{XHCV} | XTAL HIGH to CKOUT HIGH or LOW | 1A | 8 | 96 | 5 | 45 | 5 | 40 | 5 | 30 | ns |
| t_{COL2} | CKOUT, LOW level | 2 | 33 | - | 25 | - | 20 | - | 15 | - | ns |
| t_{COH} | CKOUT, HIGH level | 3 | 33 | - | 25 | - | 20 | - | 15 | - | ns |
| t_{COF} | CKOUT fall-time | 4 | - | 10 | - | 10 | - | 10 | - | 10 | ns |
| t_{COR} | CKOUT rise-time | 5 | - | 12 | - | 10 | - | 10 | - | 10 | ns |
| t_{CLAV} | CKOUT LOW to address valid | 6 | - | 55 | - | 50 | - | 50 | - | 45 | ns |
| t_{CHAZx} | CKOUT HIGH to address/data, high-impedance (max.) | 7 | - | 55 | - | 55 | - | 55 | - | 55 | ns |
| t_{CHAZn} | CKOUT HIGH to address invalid (min.) | 8 | 0 | - | 0 | - | 0 | - | 0 | - | ns |
| t_{CHSL} | CKOUT HIGH to ASN, DSN LOW | 9 | 0 | 45 | 0 | 45 | 0 | 45 | 0 | 30 | ns |
| t_{AVSL} | Address to ASN/DSN (read), ASN (write) LOW | 11 ⁸ | 20 | - | 10 | - | 10 | - | 10 | - | ns |
| t_{SLSH} | CKOUT LOW to ASN, DSN HIGH | 12 | 0 | 55 | 0 | 45 | 0 | 45 | 0 | 45 | ns |
| t_{SHAZ} | ASN, DSN HIGH to address invalid | 13 ⁸ | 20 | - | 10 | - | 10 | - | 10 | - | ns |
| t_{SL} | ASN/DSN (read), ASN (write) LOW level | 14 ⁸ | 200 | - | 160 | - | 130 | - | 105 | - | ns |
| t_{DSL} | DSN LOW level (write) | 14A ⁸ | 100 | - | 80 | - | 65 | - | 50 | - | ns |
| t_{SH} | ASN, DSN HIGH level | 15 | 100 | - | 80 | - | 70 | - | 60 | - | ns |
| t_{CHSZ} | CKOUT HIGH to ASN, DSN high impedance | 16 | - | 55 | - | 55 | - | 50 | - | 50 | ns |
| t_{SHRH} | ASN, DSN HIGH to R/WN HIGH (read) | 17 ⁸ | 20 | - | 10 | - | 10 | - | 10 | - | ns |
| t_{CHRH} | CKOUT HIGH to R/WN HIGH | 18 | 0 | 55 | 0 | 45 | 0 | 45 | 0 | 45 | ns |
| t_{CHRL} | CKOUT HIGH to R/WN LOW (write) | 20 | - | 55 | - | 45 | - | 45 | - | 45 | ns |
| t_{AVRL} | Address valid to R/WN LOW (write) | 21 ⁸ | 0 | - | 0 | - | 0 | - | 0 | - | ns |
| t_{CLSL} | R/WN LOW to DSN LOW (write) | 22 ⁸ | 55 | - | 30 | - | 25 | - | 20 | - | ns |
| t_{CLDO} | CKOUT LOW to data out valid (write) | 23 | - | 50 | - | 45 | - | 45 | - | 45 | ns |

Note. 'cp' denotes clock pulses.

16/32-bit microprocessor

SCC68070

| SYMBOL | PARAMETER | NO. | f _{XTAL1} =19.6 MHz | | f _{XTAL1} =25 MHz | | f _{XTAL1} =30 MHz | | f _{XTAL1} =35 MHz | | UNIT |
|---------------------|--|--------------------|------------------------------|------|----------------------------|------|----------------------------|------|----------------------------|------|----------|
| | | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | |
| t _{SHDO} | ASN, DSN HIGH to data out invalid (write) | 25 ⁸ | 20 | - | 15 | - | 15 | - | 15 | - | ns |
| t _{DOSL} | Data out valid to DSN LOW (write) | 26 ⁸ | 20 | - | 15 | - | 15 | - | 15 | - | ns |
| t _{DICL} | Data in to clock LOW (set-up time, read) | 27 ¹⁰ | 10 | - | 10 | - | 5 | - | 5 | - | ns |
| t _{SHDAH} | ASN, DSN HIGH to DTACKN, RDYN, AVN HIGH | 28 ⁸ | 0 | 190 | 0 | 150 | 0 | 120 | 0 | 90 | ns |
| t _{SHDI} | ASN, DSN HIGH to data invalid (hold time, read) | 29 | 0 | - | 0 | - | 0 | - | 0 | - | ns |
| t _{SHBEH} | ASN, DSN HIGH to BERRN HIGH | 30 | 0 | - | 0 | - | 0 | - | 0 | - | ns |
| t _{DCLDI} | DTACKN LOW to data in (set-up time, read) | 31 ^{8/10} | - | 65 | - | 50 | - | 45 | - | 40 | ns |
| t _{RHf} | HALTN and RESETN input transition time | 32 ⁹ | 0 | 200 | 0 | 200 | 0 | 200 | 0 | 200 | ns |
| t _{CHGL} | CKOUT HIGH to BGN LOW | 33 | - | 55 | - | 50 | - | 50 | - | 50 | ns |
| t _{CHGH} | CKOUT HIGH to BGN HIGH | 34 | - | 55 | - | 50 | - | 50 | - | 50 | ns |
| t _{BRLGL} | BRN LOW to BGN LOW | 35 | 1.5 | 3.5 | 1.5 | 3.5 | 1.5 | 3.5 | 1.5 | 3.5 | cp ns |
| t _{BRHGH} | BRN HIGH to BGN HIGH | 36 | 1.5 | 2.5 | 1.5 | 2.5 | 1.5 | 2.5 | 1.5 | 2.5 | cp ns |
| t _{GALGH} | BGACKN LOW to BGN HIGH | 37 | 1.5 | 2.5 | 1.5 | 2.5 | 1.5 | 2.5 | 1.5 | 2.5 | cp ns |
| t _{GLZ} | BGN LOW to bus high impedance (ASN HIGH) | 38 | - | 55 | - | 50 | - | 50 | - | 50 | ns |
| t _{GH} | BGN HIGH level | 39 | 1.5 | - | 1.5 | - | 1.5 | - | 1.5 | - | cp |
| t _{BGL} | BGACKN width | 40 | 1.5 | - | 1.5 | - | 1.5 | - | 1.5 | - | cp |
| t _{ASI} | Asynchronous set-up time | 41 | 25 | - | 25 | - | 25 | - | 25 | - | ns |
| t _{ASDT} | Asynchronous set-up time for DTACKN, AVN, BERRN, HALTN, RDYN | 41A ¹⁰ | 25 | - | 10 | - | 10 | - | 10 | - | ns |
| t _{BELDAL} | BERRN (input) LOW to DTACKN LOW | 42 ¹¹ | 20 | - | 15 | - | 15 | - | 15 | - | ns |
| t _{CHDO} | CKOUT HIGH to Data OUT invalid (write) | 43 | 0 | - | 0 | - | 0 | - | 0 | - | ns |
| t _{RLDL} | R/WN LOW to data bus driven | 44 | 20 | - | 10 | - | 10 | - | 10 | - | ns |

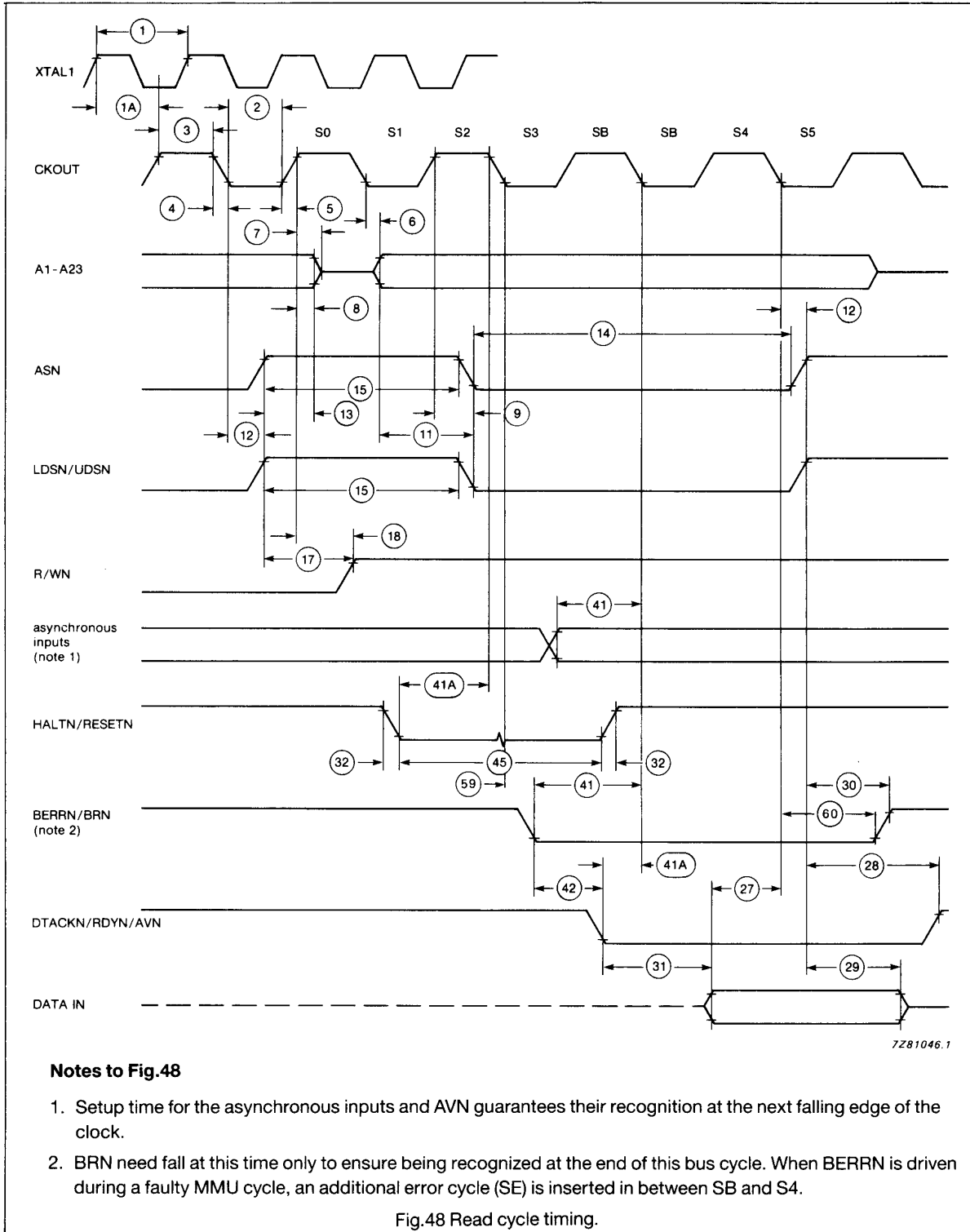
16/32-bit microprocessor

SCC68070

| SYMBOL | PARAMETER | NO. | f _{XTAL1} =19.6 MHz | | f _{XTAL1} =25 MHz | | f _{XTAL1} =30 MHz | | f _{XTAL1} =35 MHz | | UNIT |
|-------------------|---|-----|------------------------------|------|----------------------------|------|----------------------------|------|----------------------------|------|------|
| | | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | |
| t _{HRPW} | HALTN/RESETN pulse width | 45 | 10 | - | 10 | - | 10 | - | 10 | - | cp |
| | REQx set-up before CKOUT LOW | 46 | 25 | - | 10 | - | 10 | - | 10 | - | ns |
| | ACKxN LOW from CKOUT HIGH | 47 | 0 | 50 | 0 | 50 | 0 | 50 | 0 | 50 | ns |
| | REQxN hold after CKOUT LOW | 48 | 10 | - | 10 | - | 10 | - | 10 | - | ns |
| | DTCN LOW from CKOUT HIGH | 49 | - | 50 | - | 50 | - | 50 | - | 50 | ns |
| | ASN, LDSN, UDSN HIGH from DTCN LOW | 50 | 0 | - | 0 | - | 10 | - | 10 | - | ns |
| | ACKxN HIGH from CKOUT HIGH | 51 | - | 55 | - | 50 | - | 50 | - | 50 | ns |
| | DTCN non-active to CKOUT HIGH | 52 | - | 45 | - | 40 | - | 40 | - | 40 | ns |
| | DONEN (output) LOW from CKOUT HIGH | 53 | - | 45 | - | 40 | - | 40 | - | 40 | ns |
| | DONEN (output) non-active from CKOUT HIGH | 54 | - | 55 | - | 55 | - | 55 | - | 55 | ns |
| | DONEN (input) set-up LOW before CKOUT LOW | 56 | 25 | - | 10 | - | 10 | - | 10 | - | ns |
| | DONEN (input) hold LOW after CKOUT HIGH | 57 | 10 | - | 10 | - | 10 | - | 10 | - | ns |
| | REQ LOW to BGACKN (output) LOW | 58 | 3.5 | - | 3.5 | - | 3.5 | - | 3.5 | - | cp |
| | CKOUT LOW to BERRN (output) LOW | 59 | - | 50 | - | 40 | - | 40 | - | 40 | ns |
| | CKOUT LOW to BERRN (output) non-active | 60 | - | 45 | - | 45 | - | 45 | - | 45 | ns |
| | CKOUT HIGH to BGACKN (output) LOW | 61 | - | 60 | - | 60 | - | 60 | - | 60 | ns |

16/32-bit microprocessor

SCC68070



16/32-bit microprocessor

SCC68070

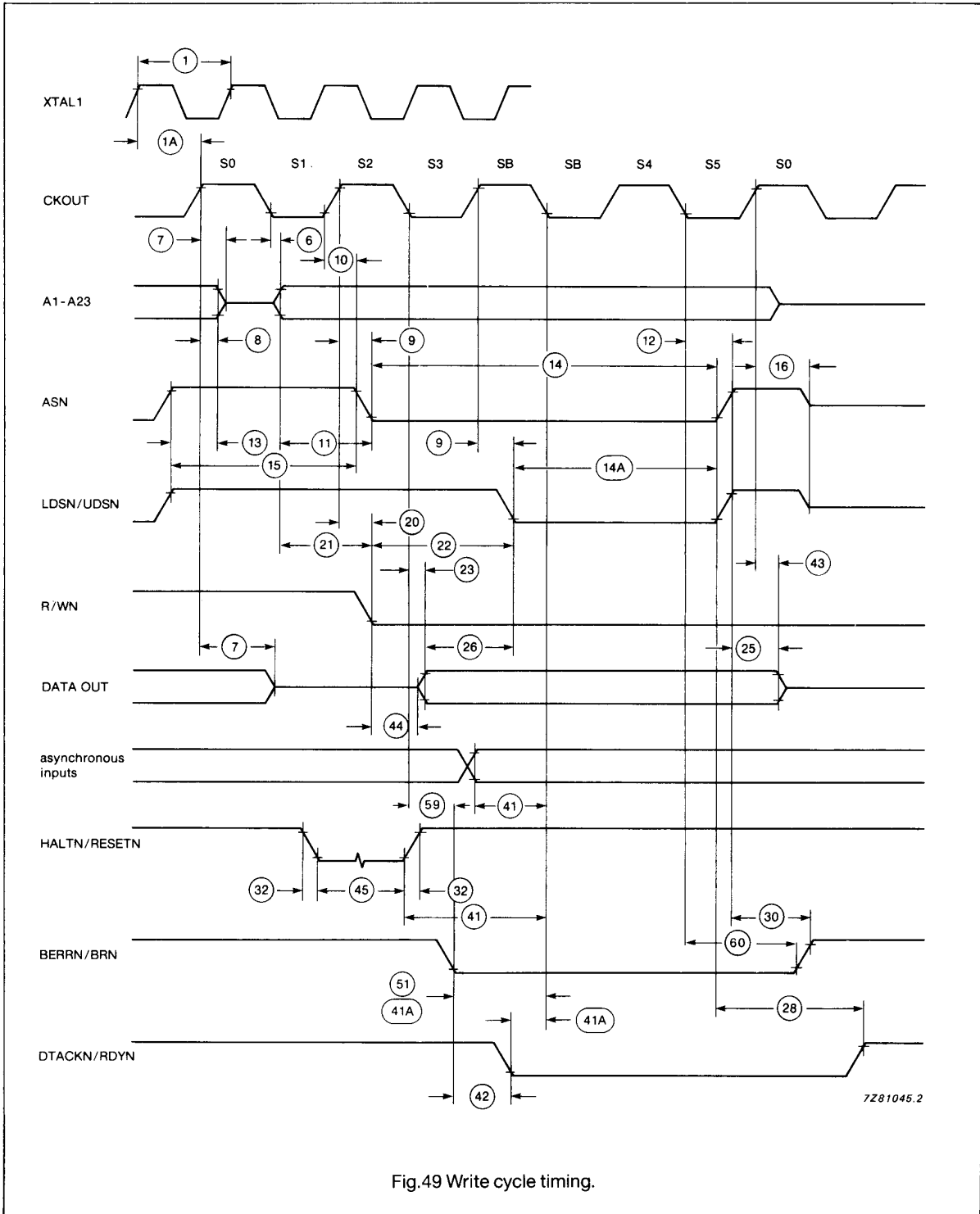


Fig.49 Write cycle timing.

7Z81045.2

16/32-bit microprocessor

SCC68070

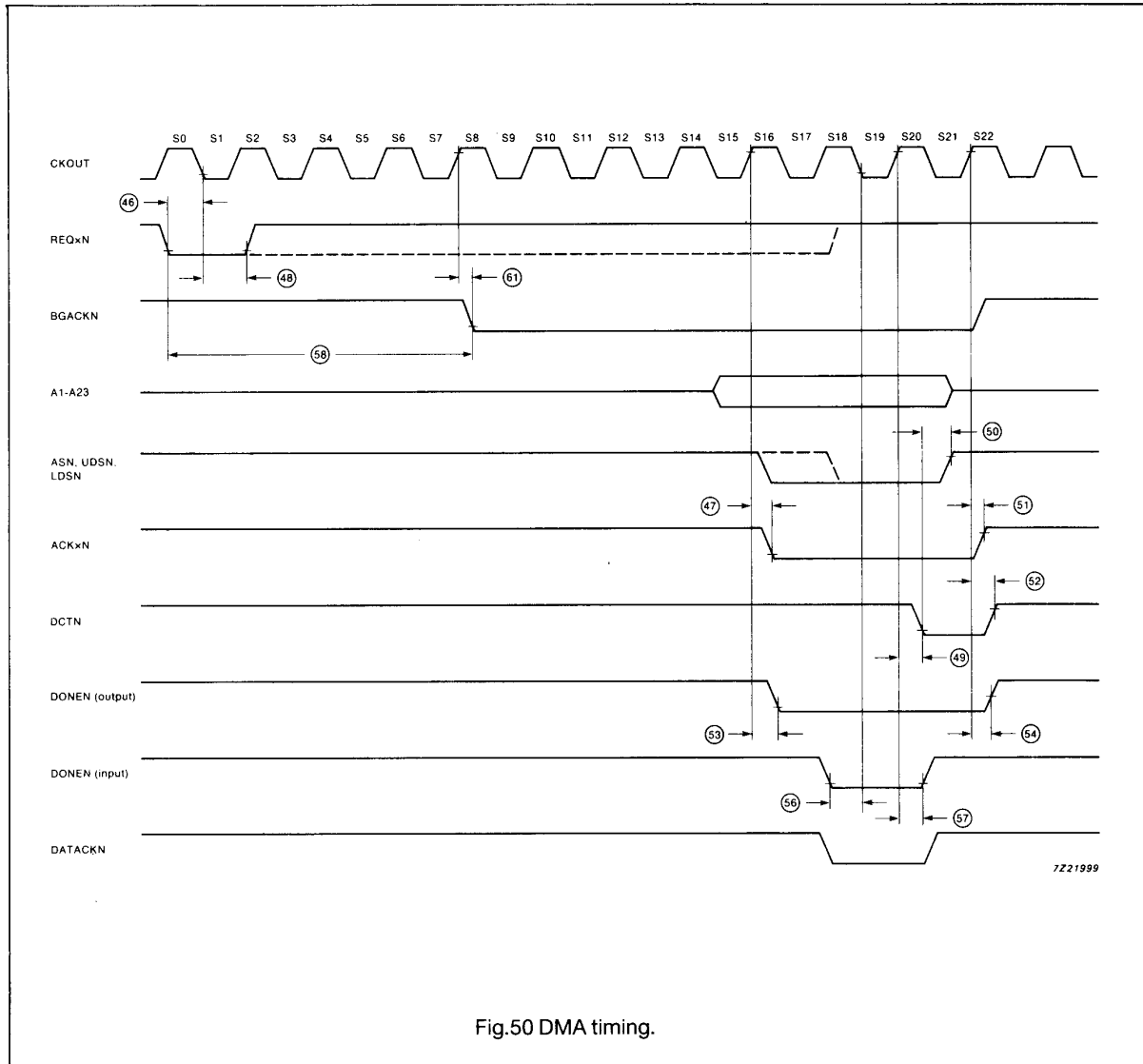
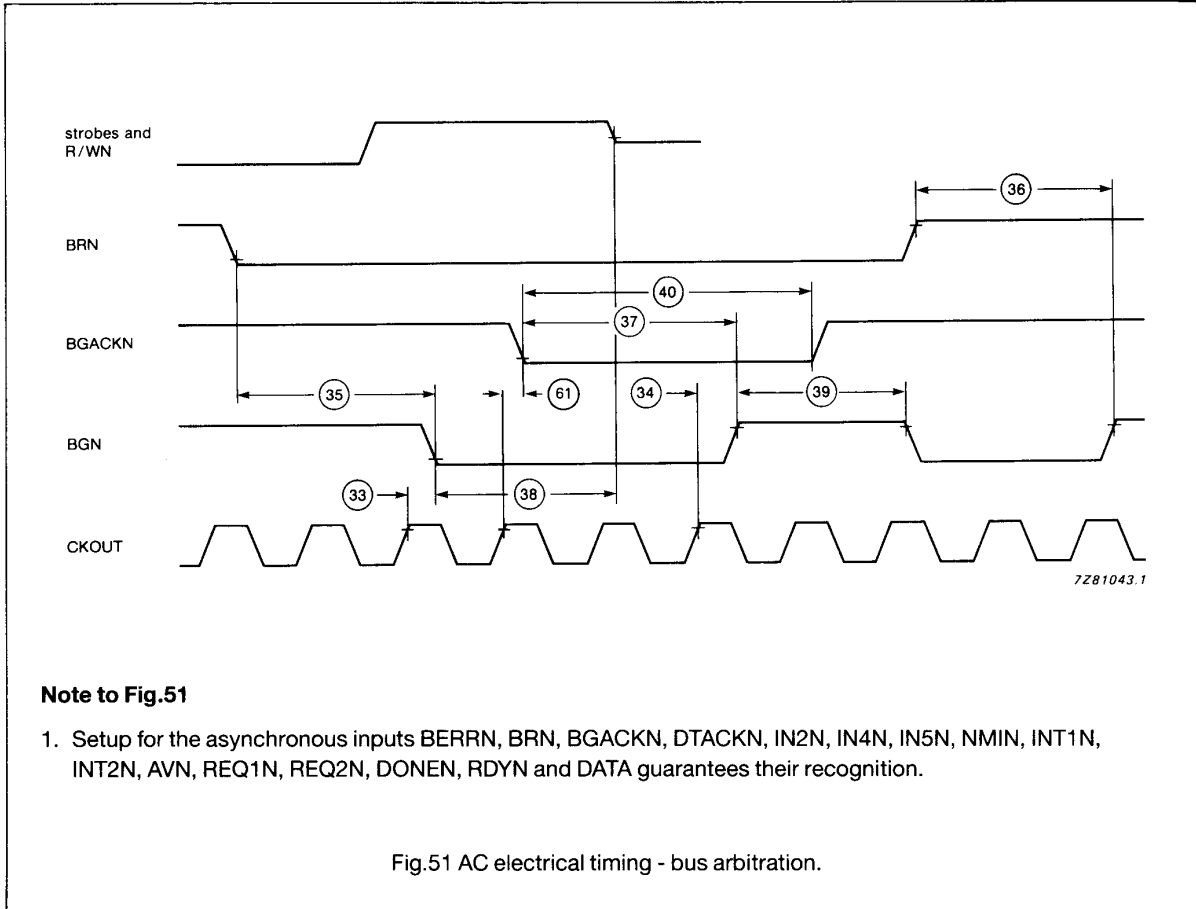


Fig.50 DMA timing.

16/32-bit microprocessor

SCC68070



16/32-bit microprocessor

SCC68070

POWER CONSIDERATIONS

The average chip-junction temperature T_j , in °C can be obtained from:

$$T_j = T_{amb} + (P_d \times R_{thj-a}) \quad (1)$$

where :

- T_{amb} = ambient temperature (°C)
- R_{thj-a} = package thermal resistance, junction-to-ambient, (K/W)
- $P_d = P_{INT} + P_{I/O} \quad (2)$
- $P_{INT} = I_{DD} \times V_{DD} =$ chip internal power (W)
- $P_{I/O} =$ power dissipation on input and output pins (determined by the user)

For most applications $P_{I/O} < P_{INT}$. The approximate relationship between P_d and T_j (if $P_{I/O}$ is neglected) is:

$$P_d = K / (T_j + 273)$$

Solving equations (1) and (2) for K gives:

$$K = P_d(T_{amb} + 273) + R_{thj-a}P_d^2 \quad (3)$$

Where K is a constant pertaining to a particular part. K can be determined from equation (3) by measuring P_d (at equilibrium) for a known T_{amb} . Using this value of K, the values of P_d and T_j can be obtained by solving equations (1) and (2) for any value of T_{amb} .

CLOCK TIMING (see Figs. 52 and 53)

Table 25 Clock timing.

| SYMBOL | PARAMETER | MIN. | MAX. | UNIT |
|------------------|----------------------------|------|------|------|
| f | Crystal or input frequency | 8 | 35 | MHz |
| t _{cyc} | Cycle time | 28.5 | 125 | ns |
| t _{CL} | Clock pulse width | 10 | - | ns |
| t _{CH} | | 10 | - | ns |
| t _{Cr} | Rise and fall times | - | 10 | ns |
| t _{Cf} | | - | 10 | ns |
| t _{C12} | XTAL1 HIGH to XTAL2 LOW | - | 10 | ns |
| t _{C21} | XTAL1 LOW to XTAL2 HIGH | - | 10 | ns |

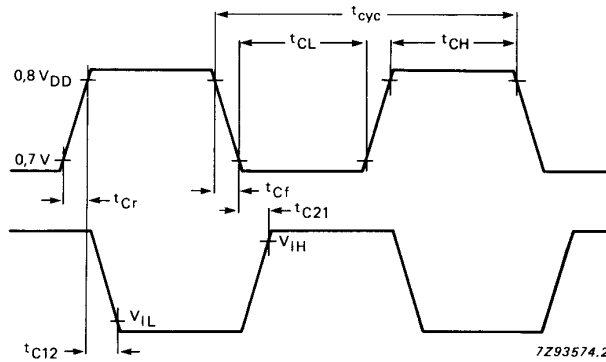


Fig.52 Input clock timing when external clock used.

16/32-bit microprocessor

SCC68070

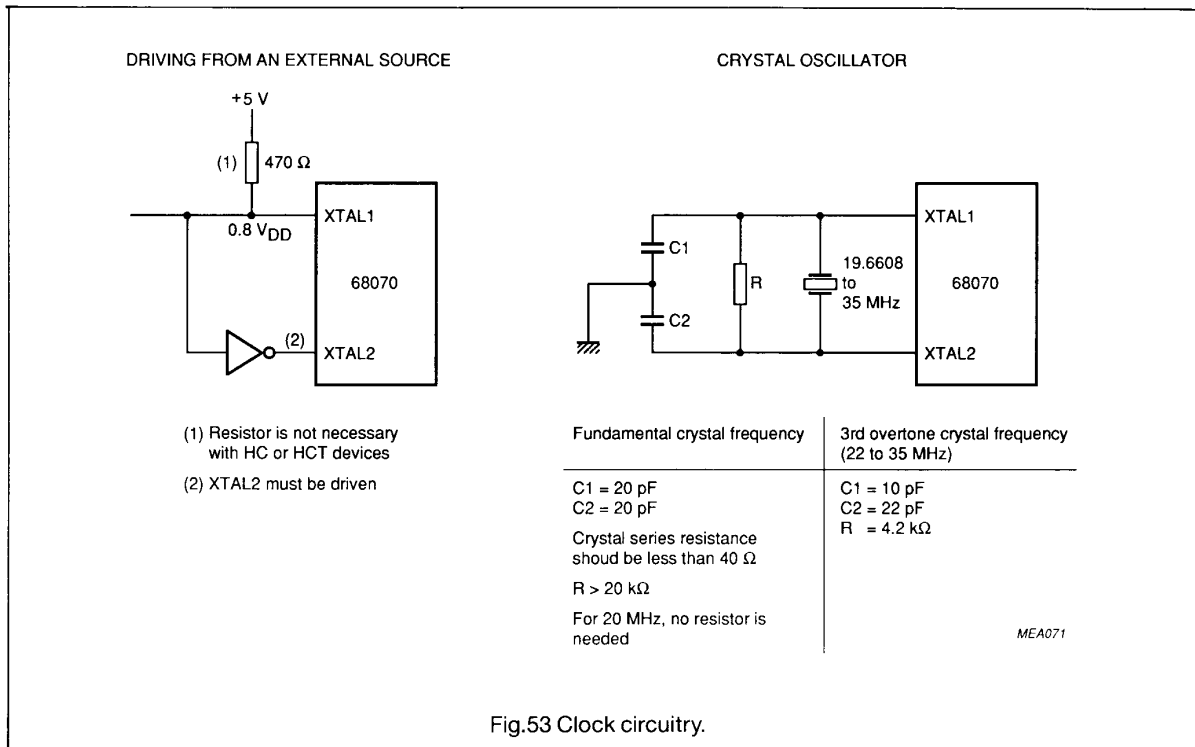


Fig.53 Clock circuitry.

I²C INTERFACE TIMING

Table 26 I²C interface timing.

| SYMBOL | PARAMETER | MIN. | MAX. | UNIT |
|---------------------|---|------|------|------|
| f _{SCL} | SCL clock frequency | 0 | 224 | KHz |
| t _{BUF} | Time the bus must be free before new transmission can start | 4.7 | - | μs |
| t _{HD;STA} | Hold time START condition. After this period the first clock pulse is generated | 4.0 | - | μs |
| t _{LOW} | LOW period of clock | 4.7 | - | μs |
| t _{HIGH} | HIGH period of clock | 4.0 | - | μs |
| t _{SU;STA} | Set-up time for START condition (only relevant for a repeated start condition) | 4.7 | - | μs |
| t _{HD;DAT} | Hold time DATA for I ² C devices | 0 | - | μs |
| t _{SU;DAT} | Set-up time DATA | 250 | - | ns |
| t _R | Rise time of both SDA and SCL lines | - | 1 | μs |
| t _F | Fall time of both SDA and SCL lines | - | 300 | ns |
| t _{SO;STO} | Set-up time for stop condition | 4.7 | - | μs |

Notes to Table 26

1. All values are referenced to V_{IH} and V_{IL} levels.
2. Timings given above are for SCL = 100 kHz (maximum I²C system frequency).

16/32-bit microprocessor

SCC68070

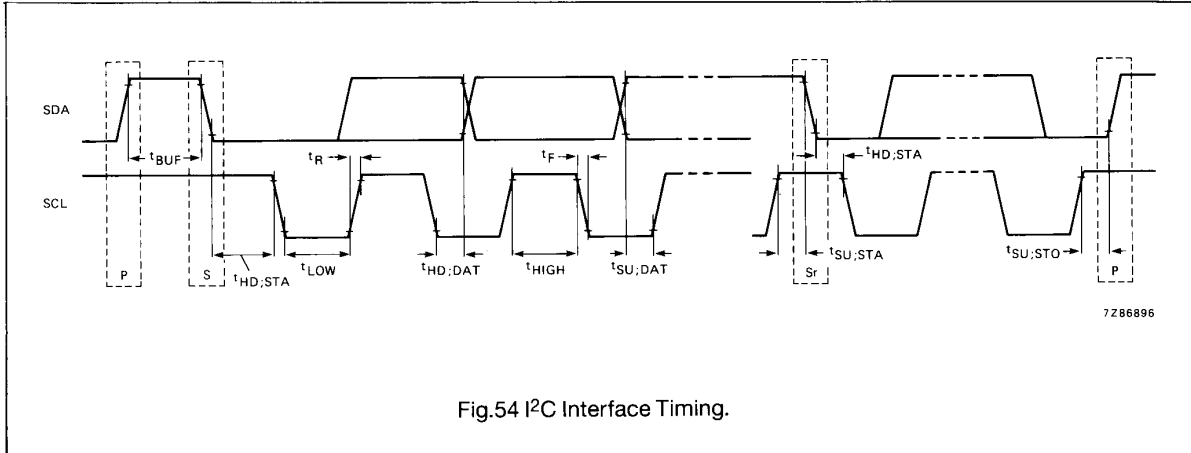


Fig.54 I²C Interface Timing.

UART INTERFACE TIMING

Table 27 UART interface timing.

| SYMBOL | PARAMETER | 10 MHZ | | 12.5 MHZ | | 15/17.5 MHZ | | UNIT |
|------------|-----------------------------|--------|------|----------|------|-------------|------|------|
| | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | |
| f_{XCKI} | XCKI frequency of operation | 2 | 5 | 2 | 10 | 2 | 10 | MHz |
| t_{Xl} | XCKI cycle time | 200 | 500 | 100 | 500 | 100 | 500 | ns |
| t_{xH} | XCKI pulse width | 60 | 250 | 30 | 250 | 30 | 250 | ns |
| t_{xL} | | 60 | 250 | 30 | 250 | 30 | 250 | ns |
| t_{xr} | XCKI rise and fall times | - | 10 | - | 10 | - | 10 | ns |
| t_{xf} | | - | 10 | - | 10 | - | 10 | ns |

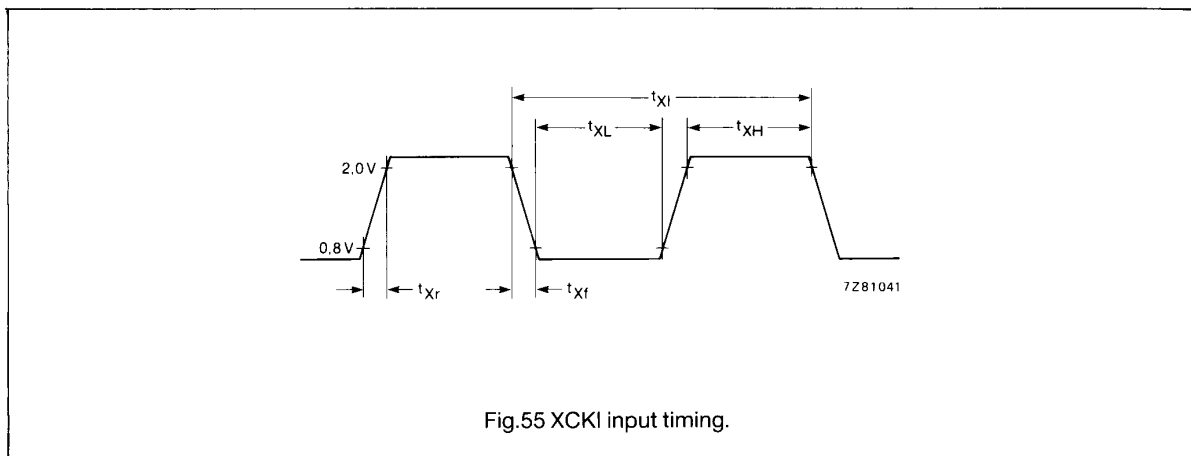


Fig.55 XCKI input timing.

16/32-bit microprocessor

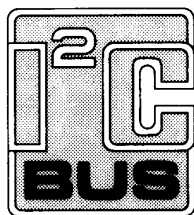
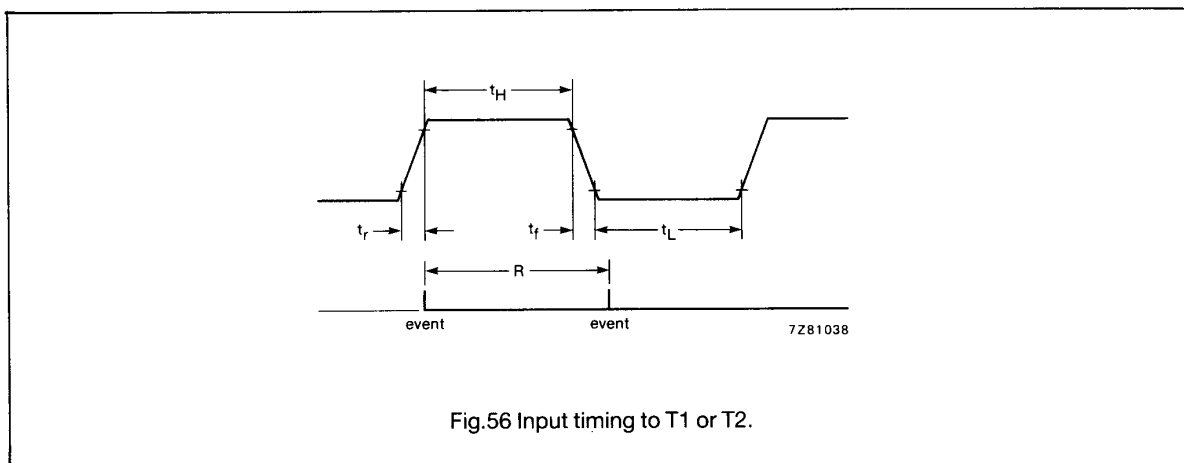
SCC68070

TIMER SPECIFICATION

Table 28 Input timing to T1 or T2.

| SYMBOL | PARAMETER | 10 MHz | | 12.5 MHz | | 15 MHz | | 17.5 MHz | | UNIT |
|--------|---|--------|------|----------|------|--------|------|----------|------|---------|
| | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | |
| t_H | T1 or T2 pulse width | 700 | - | 560 | - | 466 | - | 400 | - | ns |
| t_L | | 700 | - | 560 | - | 466 | - | 400 | - | ns |
| t_r | T1 or T2 rise and fall times | - | - | - | - | - | 50 | - | 50 | ns |
| t_f | | - | - | - | - | - | 50 | - | 50 | ns |
| R | Resolution (the time between two events to be taken into account) | 9.6 | - | 7.68 | - | 6.4 | - | 5.48 | - | μ s |

T1 and T2 input signals must be held HIGH or LOW longer than t_H or t_L to be latched at the input to the Timer. Events must be separated by more than the resolution R of the Timer.



Purchase of Philips' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C-system provided the system conforms to the I²C specifications defined by Philips.

16/32-bit microprocessor**SCC68070****SUMMARY OF ON-CHIP ADDRESSES**

| HEX ADDRESS | SYMBOL | REGISTER |
|------------------------|--------|---|
| 8000 0000 to 8000 1000 | - | reserved |
| 8000 1001 | LIR | Latched Interrupt Priority Register |
| 8000 1002 to 8000 2000 | - | reserved |
| 8000 2001 | IDR | I ² C Data Register |
| 8000 2002 | - | reserved |
| 8000 2003 | IAR | I ² C Address Register |
| 8000 2004 | - | reserved |
| 8000 2005 | ISR | I ² C Status Register |
| 8000 2006 | - | reserved |
| 8000 2007 | ICR | I ² C Control Register |
| 8000 2008 | - | reserved |
| 8000 2009 | ICC | I ² C Clock Control Register |
| 8000 200A to 8000 2010 | - | reserved |
| 8000 2011 | UMR | UART Mode Register |
| 8000 2012 | - | reserved |
| 8000 2013 | USR | UART Status Register |
| 8000 2014 | - | reserved |
| 8000 2015 | UCS | UART Clock Select Register |
| 8000 2016 | - | reserved |
| 8000 2017 | UCR | UART Command Register |
| 8000 2018 | - | reserved |
| 8000 2019 | UTH | UART Transmit Holding Register |
| 8000 201A | - | reserved |
| 8000 201B | URH | UART Receive Holding Register |
| 8000 201C to 8000 201F | - | reserved |
| 8000 2020 | TSR | Timer Status Register |
| 8000 2021 | TCR | Timer Control Register |
| 8000 2022 | RRH | Reload Register High |
| 8000 2023 | RRL | Reload Register Low |
| 8000 2024 | T0H | Timer 0 High |
| 8000 2025 | T0L | Timer 0 Low |
| 8000 2026 | T1H | Timer 1 High |
| 8000 2027 | T1L | Timer 1 Low |
| 8000 2028 | T2H | Timer 2 High |
| 8000 2029 | T2L | Timer 2 Low |
| 8000 202A to 8000 2044 | - | reserved |
| 8000 2045 | PICR1 | Peripheral Interrupt Control Register 1 |
| 8000 2046 | - | reserved |
| 8000 2047 | PICR2 | Peripheral Interrupt Control Register 2 |
| 8000 2048 to 8000 3FFF | - | reserved |
| 8000 4000 | CSR | Channel Status Register Channel 1 |
| 8000 4001 | CER | Channel Error Register Channel 1 |
| 8000 4002 to 8000 4003 | - | reserved |

16/32-bit microprocessor**SCC68070**

| HEX ADDRESS | SYMBOL | REGISTER |
|------------------------|--------|--|
| 8000 4004 | DCR | Device Control Register Channel 1 |
| 8000 4005 | OCR | Operation Control Register Channel 1 |
| 8000 4006 | SCR | Sequence Control Register Channel 1 |
| 8000 4007 | CCR | Channel Control Register Channel 1 |
| 8000 4008 to 8000 4009 | - | reserved |
| 8000 400A | MTCH | Memory Transfer Counter High Channel 1 |
| 8000 400B | MTCL | Memory Transfer Counter Low Channel 1 |
| 8000 400C | MACH | Memory Address Counter High Channel 1 |
| 8000 400D | MACMH | Memory Address Counter Middle High Channel 1 |
| 8000 400E | MACML | Memory Address Counter Middle Low Channel 1 |
| 8000 400F | MACL | Memory Address Counter Low Channel 1 |
| 8000 4010 to 8000 402C | - | reserved |
| 8000 402D | CPR | Channel Priority Register Channel 1 |
| 8000 402E to 8000 403F | - | reserved |
| 8000 4040 | CSR | Channel Status Register Channel 2 |
| 8000 4041 | CER | Channel Error Register Channel 2 |
| 8000 4042 to 8000 4043 | - | reserved |
| 8000 4044 | DCR | Device Control Register Channel 2 |
| 8000 4045 | OCR | Operation Control Register Channel 2 |
| 8000 4046 | SCR | Sequence Control Register Channel 2 |
| 8000 4047 | CCR | Channel Control Register Channel 2 |
| 8000 4048 to 8000 4049 | - | reserved |
| 8000 404A | MTCH | Memory Transfer Counter High Channel 2 |
| 8000 404B | MTCL | Memory Transfer Counter Low Channel 2 |
| 8000 404C | MACH | Memory Address Counter High Channel 2 |
| 8000 404D | MACMH | Memory Address Counter Middle High Channel 2 |
| 8000 404E | MACML | Memory Address Counter Middle Low Channel 2 |
| 8000 404F | MACL | Memory Address Counter Low Channel 2 |
| 8000 4050 to 8000 4053 | - | reserved |
| 8000 4054 | DACH | Device Address Counter High Channel 2 |
| 8000 4055 | DACHMH | Device Address Counter Middle High Channel 2 |
| 8000 4056 | DACML | Device Address Counter Middle Low Channel 2 |
| 8000 4057 | DACL | Device Address Counter Low Channel 2 |
| 8000 4058 to 8000 406C | - | reserved |
| 8000 406D | CPR | Channel Priority Register Channel 2 |
| 8000 406E to 8000 7FFF | - | reserved |
| 8000 8000 | MSR | MMU Status Register |
| 8000 8001 | MCR | MMU Control Register |
| 8000 8002 to 8000 803F | - | reserved |
| 8000 8040 | SAH | Segment Attributes High, Descriptor 0 |
| 8000 8041 | SAL | Segment Attributes Low, Descriptor 0 |
| 8000 8042 | SLH | Segment Length High, Descriptor 0 |
| 8000 8043 | SLL | Segment Length Low, Descriptor 0 |
| 8000 8044 | - | reserved |
| 8000 8045 | SNR | Segment Number, Descriptor 0 |

16/32-bit microprocessor**SCC68070**

| HEX ADDRESS | SYMBOL | REGISTER |
|-------------|--------|---|
| 8000 8046 | SBH | Segment Base Address High, Descriptor 0 |
| 8000 8047 | SBL | Segment Base Address Low, Descriptor 0 |
| 8000 8048 | SAH | Segment Attributes High, Descriptor 1 |
| 8000 8049 | SAL | Segment Attributes Low, Descriptor 1 |
| 8000 804A | SLH | Segment Length High, Descriptor 1 |
| 8000 804B | SLL | Segment Length Low, Descriptor 1 |
| 8000 804C | - | reserved |
| 8000 804D | SNR | Segment Number, Descriptor 1 |
| 8000 804E | SBH | Segment Base Address High, Descriptor 1 |
| 8000 804F | SBL | Segment Base Address Low, Descriptor 1 |
| 8000 8050 | SAH | Segment Attributes High, Descriptor 2 |
| 8000 8051 | SAL | Segment Attributes Low, Descriptor 2 |
| 8000 8052 | SLH | Segment Length High, Descriptor 2 |
| 8000 8053 | SLL | Segment Length Low, Descriptor 2 |
| 8000 8054 | - | reserved |
| 8000 8055 | SNR | Segment Number, Descriptor 2 |
| 8000 8056 | SBH | Segment Base Address High, Descriptor 2 |
| 8000 8057 | SBL | Segment Base Address Low, Descriptor 2 |
| 8000 8058 | SAH | Segment Attributes High, Descriptor 3 |
| 8000 8059 | SAL | Segment Attributes Low, Descriptor 3 |
| 8000 805A | SLH | Segment Length High, Descriptor 3 |
| 8000 805B | SLL | Segment Length Low, Descriptor 3 |
| 8000 805C | - | reserved |
| 8000 805D | SNR | Segment Number, Descriptor 3 |
| 8000 805E | SBH | Segment Base Address High, Descriptor 3 |
| 8000 805F | SBL | Segment Base Address Low, Descriptor 3 |
| 8000 8060 | SAH | Segment Attributes High, Descriptor 4 |
| 8000 8061 | SAL | Segment Attributes Low, Descriptor 4 |
| 8000 8062 | SLH | Segment Length High, Descriptor 4 |
| 8000 8063 | SLL | Segment Length Low, Descriptor 4 |
| 8000 8064 | - | reserved |
| 8000 8065 | SNR | Segment Number, Descriptor 4 |
| 8000 8066 | SBH | Segment Base Address High, Descriptor 4 |
| 8000 8067 | SBL | Segment Base Address Low, Descriptor 4 |
| 8000 8068 | SAH | Segment Attributes High, Descriptor 5 |
| 8000 8069 | SAL | Segment Attributes Low, Descriptor 5 |
| 8000 806A | SLH | Segment Length High, Descriptor 5 |
| 8000 806B | SLL | Segment Length Low, Descriptor 5 |
| 8000 806C | - | reserved |
| 8000 806D | SNR | Segment Number, Descriptor 5 |
| 8000 806E | SBH | Segment Base Address High, Descriptor 5 |
| 8000 806F | SBL | Segment Base Address Low, Descriptor 5 |

16/32-bit microprocessor**SCC68070**

| HEX ADDRESS | SYMBOL | REGISTER |
|------------------------|--------|---|
| 8000 8070 | SAH | Segment Attributes High, Descriptor 6 |
| 8000 8071 | SAL | Segment Attributes Low, Descriptor 6 |
| 8000 8072 | SLH | Segment Length High, Descriptor 6 |
| 8000 8073 | SLL | Segment Length Low, Descriptor 6 |
| 8000 8074 | - | reserved |
| 8000 8075 | SNR | Segment Number, Descriptor 6 |
| 8000 8076 | SBH | Segment Base Address High, Descriptor 6 |
| 8000 8077 | SBL | Segment Base Address Low, Descriptor 6 |
| 8000 8078 | SAH | Segment Attributes High, Descriptor 7 |
| 8000 8079 | SAL | Segment Attributes Low, Descriptor 7 |
| 8000 807A | SLH | Segment Length High, Descriptor 7 |
| 8000 807B | SLL | Segment Length Low, Descriptor 7 |
| 8000 807C | - | reserved |
| 8000 807D | SNR | Segment Number, Descriptor 7 |
| 8000 807E | SBH | Segment Base Address High, Descriptor 7 |
| 8000 807F | SBL | Segment Base Address Low, Descriptor 7 |
| 8000 8080 to BFFF FFFF | - | reserved |

16/32-bit microprocessor

SCC68070

PACKAGE OUTLINES

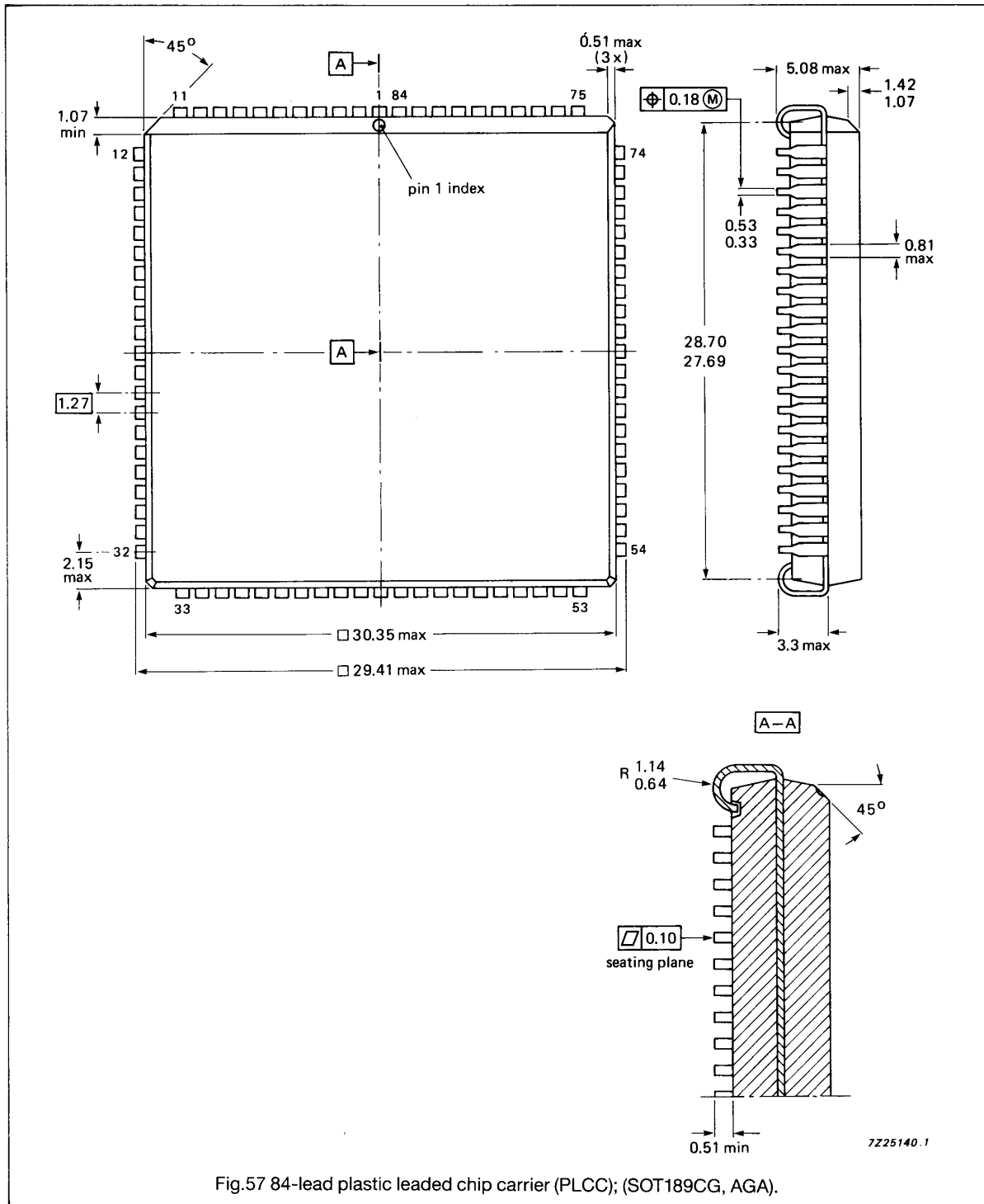
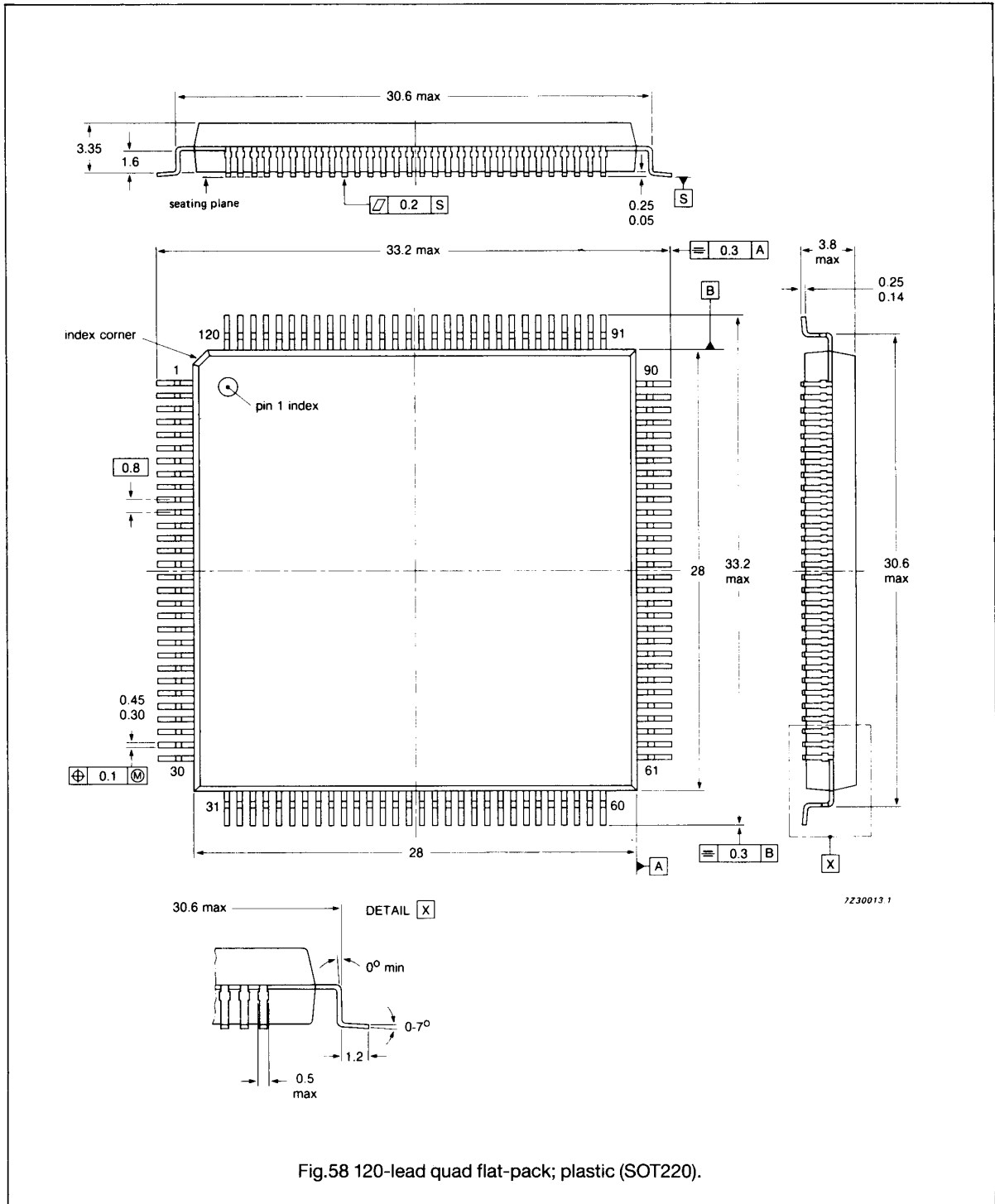


Fig.57 84-lead plastic leaded chip carrier (PLCC); (SOT189CG, AGA).

16/32-bit microprocessor

SCC68070



16/32-bit microprocessor**SCC68070**

SOLDERING**Plastic mini-packs****BY WAVE**

During placement and before soldering, the component must be fixed with a droplet of adhesive. After curing the adhesive, the component can be soldered. The adhesive can be applied by screen printing, pin transfer or syringe dispensing.

Maximum permissible solder temperature is 260 °C, and maximum duration of package immersion in solder bath is 10 s, if allowed to cool to less than 150 °C within 6 s. Typical dwell time is 4 s at 250 °C.

A modified wave soldering technique is recommended using two solder waves (dual-wave) in which a turbulent wave with high upward pressure is followed by a smooth laminar wave. Using a mildly-activated flux eliminates the need for removal of corrosive residues in most applications.

BY SOLDER PASTE REFLOW

Reflow soldering requires the solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the substrate by screen printing, stencilling or pressure-syringe dispensing before device placement.

Several techniques exist for reflowing; for example, thermal conduction by heated belt, infrared, and vapour-phase reflow. Dwell times vary between 50 and 300 s according to method. Typical reflow temperatures range from 215 to 250 °C.

Preheating is necessary to dry the paste and evaporate the binding agent. Preheating duration: 45 min at 45 °C.

REPAIRING SOLDERED JOINTS (BY HAND-HELD SOLDERING IRON OR PULSE-HEATED SOLDER TOOL)

Fix the component by first soldering two, diagonally opposite, end pins. Apply the heating tool to the flat part of the pin only. Contact time must be limited to 10 s at up to 300 °C. When using proper tools, all other pins can be soldered in one operation within 2 to 5 s at between 270 and 320 °C. (Pulse-heated soldering is not recommended for SO packages).

For pulse-heated solder tool (resistance) soldering of VSO packages, solder is applied to the substrate by dipping or by an extra thick tin/lead plating before package placement.

Plastic dual in-line packages**BY DIP OR WAVE**

The maximum permissible temperature of the solder is 260 °C; this temperature must not be in contact with the joint for more than 5 s. The total contact time of successive solder waves must not exceed 5 s.

The device may be mounted up to the seating plane, but the temperature of the plastic body must not exceed the specified storage maximum. If the printed-circuit board has been preheated, forced cooling may be necessary immediately after soldering to keep the temperature within the permissible limit.

REPAIRING SOLDERED JOINTS (BY HAND)

Apply the soldering iron below the seating plane (or not more than 2 mm above it). If its temperature is below 300 °C it must not be in contact for more than 10 s; if between 300 and 400 °C, for not more than 5 s.

16/32-bit microprocessor**SCC68070****DEFINITIONS**

| | |
|--|---|
| Data sheet status | |
| Objective specification | This data sheet contains target or goal specifications for product development. |
| Preliminary specification | This data sheet contains preliminary data; supplementary data may be published later. |
| Product specification | This data sheet contains final product specifications. |
| Limiting values | |
| Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of this specification is not implied. Exposure to limiting values for extended periods may affect device reliability. | |
| © Philips Export B.V. 1990 | |
| All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner. | |
| The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights. | |

Printed in The Netherlands

9397 289 80011