Part No. 32 958 044
Related doc: 1

# DCJ11 Microprocessor

Prepared by Educational Services
of
Digital Equipment Corporation

# APPENDIX A
# DC CHARACTERISTICS

## Absolute Maximum Rating

| | |
|---|---|
| Storage Temperature Range: | -65 C to +150 C |
| Active Temperature Range: | -55 C to +125 C |
| Supply Voltage: | +7.0V |
| Input or Output Voltage Applied: | Vss -0.3V<br>Vcc +0.3V |

## Electrical Characteristics

| | |
|---|---|
| Specified Temperature Range | 0 C to +70 C |
| Specified Voltage Range | +4.75V to +5.25V |
| Test Conditions | Temperature = +70 C<br>Vss = 0V<br>Vcc = +4.75V (except as noted) |

| Symbol | Parameter | Min. | Max. | Units | Test Condition |
|---|---|---|---|---|---|
| $V_{IH}$ | High level MOS input | 70% $V_{CC}$ | | V | |
| $V_{IL}$ | Low level MOS input | | 30% $V_{CC}$ | V | |
| $V_{IHT}$ | High level TTL input | 2.2 | | V | |
| $V_{ILT}$ | Low level TTL input | | 0.8 | V | |
| $I_I$ | Input leakage current except TEST inputs (note 1) | -10.0 | 10.0 | uA | $0V \leq V_I \leq V_{CC}$ |
| $I_{ILL}$ | Input current TEST inputs (note 1) | 0.1 | 5.0 | mA | $V_I = 0V$ |
| $I_{OH}$ | Output current at high level | -2.0 | | mA | $V_O = V_{CC} - 0.4V$ |
| $I_{OL}$ | Output current at low level | 2.0 | | mA | $V_O = 0.4V$ |

| Symbol | Parameter | Min. | Max. | Units | Test Condition |
|---|---|---|---|---|---|
| $I_{OHT}$ | Output current at high TTL level | -2.0 | | mA | $V_O = 2.4V$ |
| $I_{OSH}$ | High level sustainer current (note 1) | -0.2 | -0.6 | mA | $V_O = V_{cc} - 1.0V$ |
| $I_{OSL}$ | Low level sustainer current (note 1) | 0.2 | 0.6 | mA | $V_O = 1.0V$ |
| $I_{OZ}$ | Output leàkage current (notes 1,2) | -10.0 | 10.0 | uA | $0V \le V_O \le V_{cc}$ |
| $I_{CCSB}$ | Static power supply current (notes 1,3) | | 30 | mA | |
| $C_{IN}$ | Input capacitance (note 4) | | 7 | pF | |
| $C_{IO}$ | Input/output capacitance (note 4) | | 15 | pF | |
| $C_{OUT}$ | Output capacitance (note 4) | | 15 | pF | |
| $C_{MAX}$ | DCJ11 capacitance plus external capacitance | | 50 | pF | |

NOTES

1. Tested at Vcc = 5.25V.
2. Only applies in the high impedance condition.
3. With TEST1 and TEST2 asserted, all outputs open circuit, and all other inputs equal to Vcc.
4. Sampled and guaranteed, but not tested. Does not apply to TEST1 or TEST2.

SIGNAL SUMMARY

| TYPE | NAME | APPLICABLE DC TEST |
|---|---|---|
| TTL INPUT | IRQ<3:0>, $\overline{HALT}$, $\overline{PWRF}$, $\overline{EVENT}$, $\overline{PARITY}$, DV, $\overline{MISS}$, $\overline{CONT}$, $\overline{DMR}$, $\overline{INIT}$, $\overline{FPE}$ | $V_{IHT}$, $V_{ILT}$, $I_{I}$ |
| TTL OUTPUT | DAL<21:16>, AIO<3:0>, $\overline{ALE}$, $\overline{BUFCTL}$, $\overline{SCTL}$, $\overline{STRB}$, BS<1:0>, $\overline{MAP}$, $\overline{PRDC}$ | $I_{OL}$, $I_{OHT}$, $I_{OZ}$ |
| MOS INPUT | $\overline{TEST1}$, $\overline{TEST2}$ | $V_{IH}$, $V_{IL}$, $I_{ILL}$ |
| MOS OUTPUT | CLK, CLK2 | $I_{OH}$, $I_{OL}$, $I_{OZ}$ |
| TTL I/O | $\overline{ABORT}$* | $V_{ILT}$, $I_{OL}$, $I_{OHT}$, $I_{OZ}$, $I_{OSH}$ |
| TTL I/O | DAL<15:00> | $V_{IHT}$, $V_{ILT}$, $I_{OL}$, $I_{OHT}$, $I_{OZ}$ |
| Power | Vcc | $I_{CCSB}$ |

* $\overline{ABORT}$ must be driven with an open collector driver because the DCJ11 has a pull-up device that supplies $I_{OSH}$.

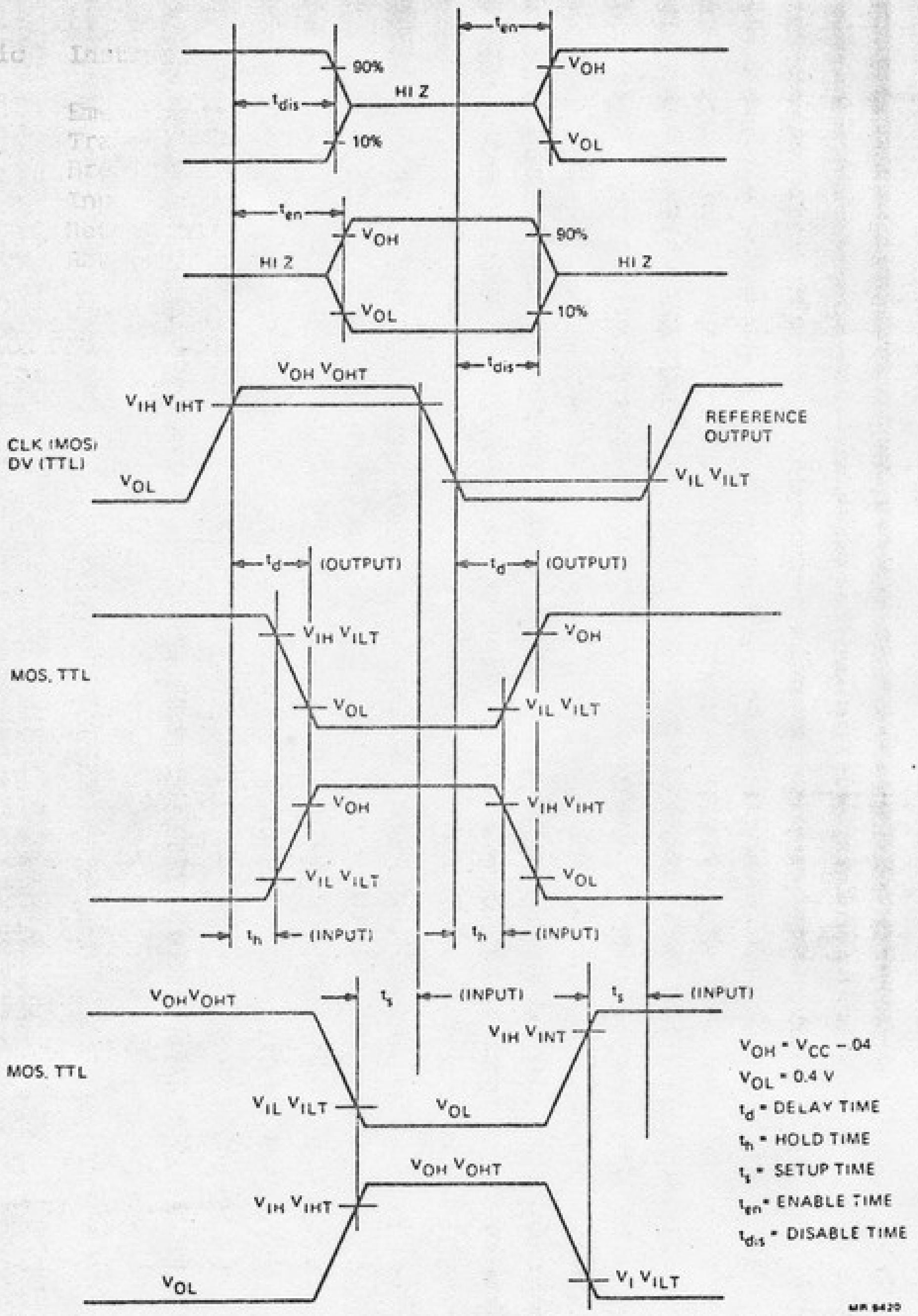


Figure A-1 Voltage Waveforms

## APPENDIX B
## AC CHARACTERISTICS

Test Conditions:

Temperature = +70 C
Vss = 0V
Vcc = +4.75V (except as noted)
CMAX = 50 pF

Timing Requirements

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $t_{INITW}$ | $\overline{\text{INIT}}$ pulse width | 10 | | clock periods |
| $t_{SCTLLH}$ | Initialization interval | 225 | | ns |
| $t_{DS}$ | DAL<15:00> setup, with respect to T3 | 35 | | ns |
| $t_{DH}$ | DAL<15:00> hold, with respect to T3 | 20 | | ns |
| $t_{DVDS}$ | DAL<15:00> setup, with respect to DV | 35 | | ns |
| $t_{DVDH}$ | DAL<15:00> hold, with respect to DV | 35 | | ns |
| $t_{DVW}$ | DV Pulse width | 35 | | ns |
| $t_{DVF}$ | DV Fall time | | 15 | ns |
| t DVH | DV deassertion with respect to T6. | 0 | | ns |
| t DVS | DV deassertion with respect to T4. | 0 | | ns |
| $t_{HMS}$ | $\overline{\text{MISS}}$ setup | 30 | | ns |
| $t_{HMH}$ | $\overline{\text{MISS}}$ hold | 10 | | ns |
| $t_{SVCS}$ | IRQ<3:0>, $\overline{\text{HALT}}$, $\overline{\text{PWRF}}$, $\overline{\text{FPE}}$, $\overline{\text{EVENT}}$ setup (see note) | 20 | | ns |

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $t_{SVCH}$ | $\overline{\text{IRQ<3:0>}}$, $\overline{\text{HALT}}$, $\overline{\text{PWRF}}$, $\overline{\text{FPE}}$, $\overline{\text{EVENT}}$ hold (see note) | 20 | | ns |
| $t_{PARS}$ | $\overline{\text{PARITY}}$ setup (see note) | 20 | | ns |
| $t_{PARH}$ | $\overline{\text{PARITY}}$ hold (see note) | 20 | | ns |
| $t_{ABS}$ | $\overline{\text{ABORT}}$ drive | 30 | | ns |
| $t_{ABD}$ | $\overline{\text{ABORT}}$ delay | 0 | | ns |
| $t_{ABW}$ | $\overline{\text{ABORT}}$ width | $40 + t_{CLKH}$ | | ns |
| $t_{CNTS}$ | $\overline{\text{CONT}}$ setup (see note) | 30 | | ns |
| $t_{CNTH}$ | $\overline{\text{CONT}}$ hold (see note) | 20 | | ns |
| $t_{DMRS}$ | $\overline{\text{DMR}}$ setup (see note) | 30 | | ns |
| $t_{DMRH}$ | $\overline{\text{DMR}}$ hold (see note) | 20 | | ns |

Note:
Setup and hold requirements are only to guarantee recognition at next sample point.

Timing Responses

| Symbol | Parameter | Min | Max | Units | Figure References |
|---|---|---|---|---|---|
| $t_{CYCLE}$ | CLK cycle time | 67 | | ns | B-1, B-4 |
| $t_{CLKH}$ | CLK high width | 28 | | ns | B-1, B-4 |
| $t_{CLKL}$ | CLK low width | 28 | | ns | B-1, B-4 |
| $t_{R}$ | CLK rise time | | 7 | ns | B-1, B-4 |
| $t_{F}$ | CLK fall time | | 7 | ns | B-1, B-4 |
| $t_{PCYC}$ | CLK2 cycle time | 67 | | ns | B-1, B-3 |
| $t_{PCLKH}$ | CLK2 high width | 28 | | ns | B-1, B-3 |

| Symbol | Parameter | Min | Max | Units | Figure References |
|---|---|---|---|---|---|
| $t_{PCLKL}$ | CLK2 low width | 28 | | ns | B-1, B-3 |
| $t_{PR}$ | CLK2 rise time | | 7 | ns | B-1, B-3 |
| $t_{PF}$ | CLK2 fall time | | 7 | ns | B-1, B-3 |
| $t_{PCLKD}$ | CLK2 valid delay | | tbs | ns | B-1, B-3 |
| $t_{MAPD}$ | MAP delay | | 45 | ns | B-1, B-3 |
| $t_{SD}$ | Strobe active delay | 0 | | ns | B-3 |
| $t_{SID}$ | Strobe inactive delay | 0 | | ns | B-3 |
| $t_{DIS}$ | DAL output disable | | 35 | ns | B-2 |
| $t_{DALD}$ | DAL valid delay | | 65 | ns | B-3 |
| $t_{DALH}$ | DAL valid hold | 0 | | ns | B-3 |
| $t_{PD}$ | PRDC valid delay | | 50 | ns | B-3 |
| $t_{PID}$ | PRDC invalid delay | | 50 | ns | B-3 |
| $t_{AIOD}$ | AIO<3:0> delay | | 75 | ns | B-3 |

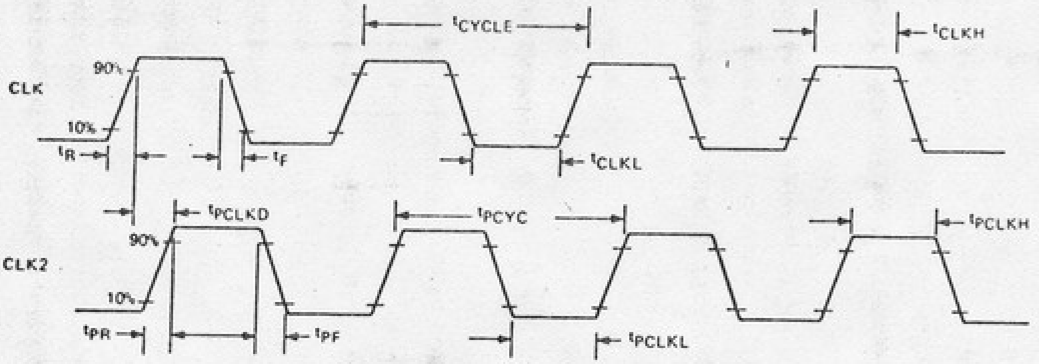


Figure B-1 Clock Timing

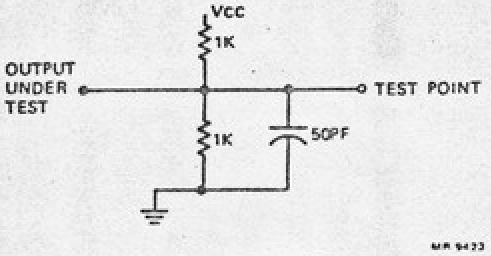


Figure B-2
Three State
Disable Test Circuit

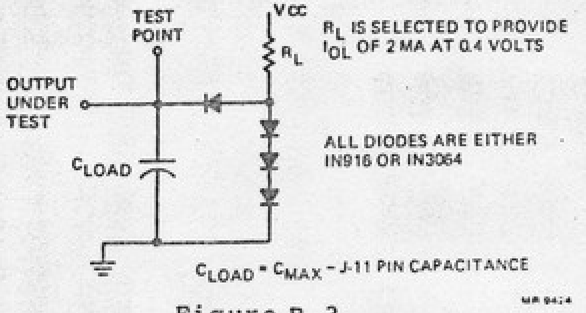

TEST POINT
VCC
RL IS SELECTED TO PROVIDE IOL OF 2 MA AT 0.4 VOLTS
RL
OUTPUT UNDER TEST
ALL DIODES ARE EITHER IN916 OR IN3064
CLOAD
CLOAD = CMAX - J-11 PIN CAPACITANCE
MR 9424

Figure B-3
TTL Output Test Circuit

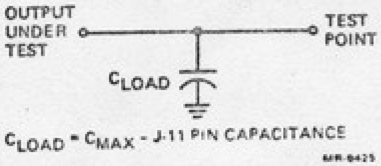


Figure B-4
MOS Output Test Circuit

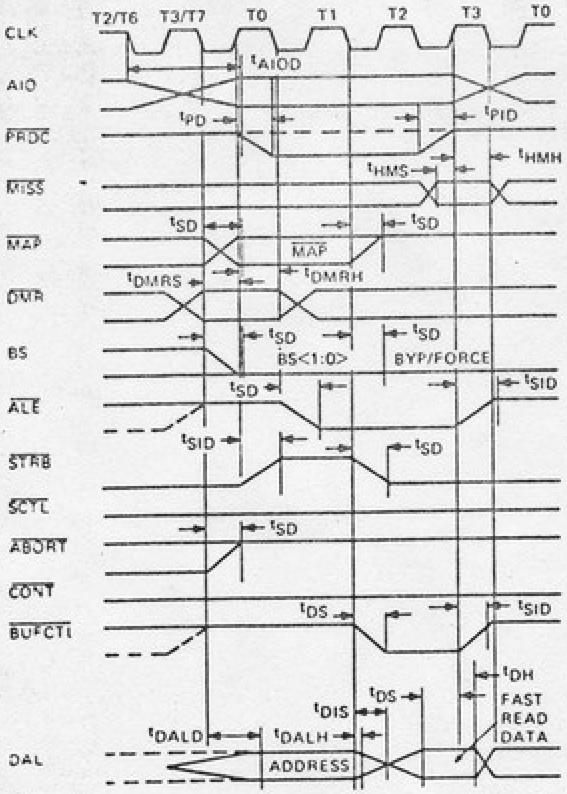


Figure B-5 Non-Stretched Bus Read Timing

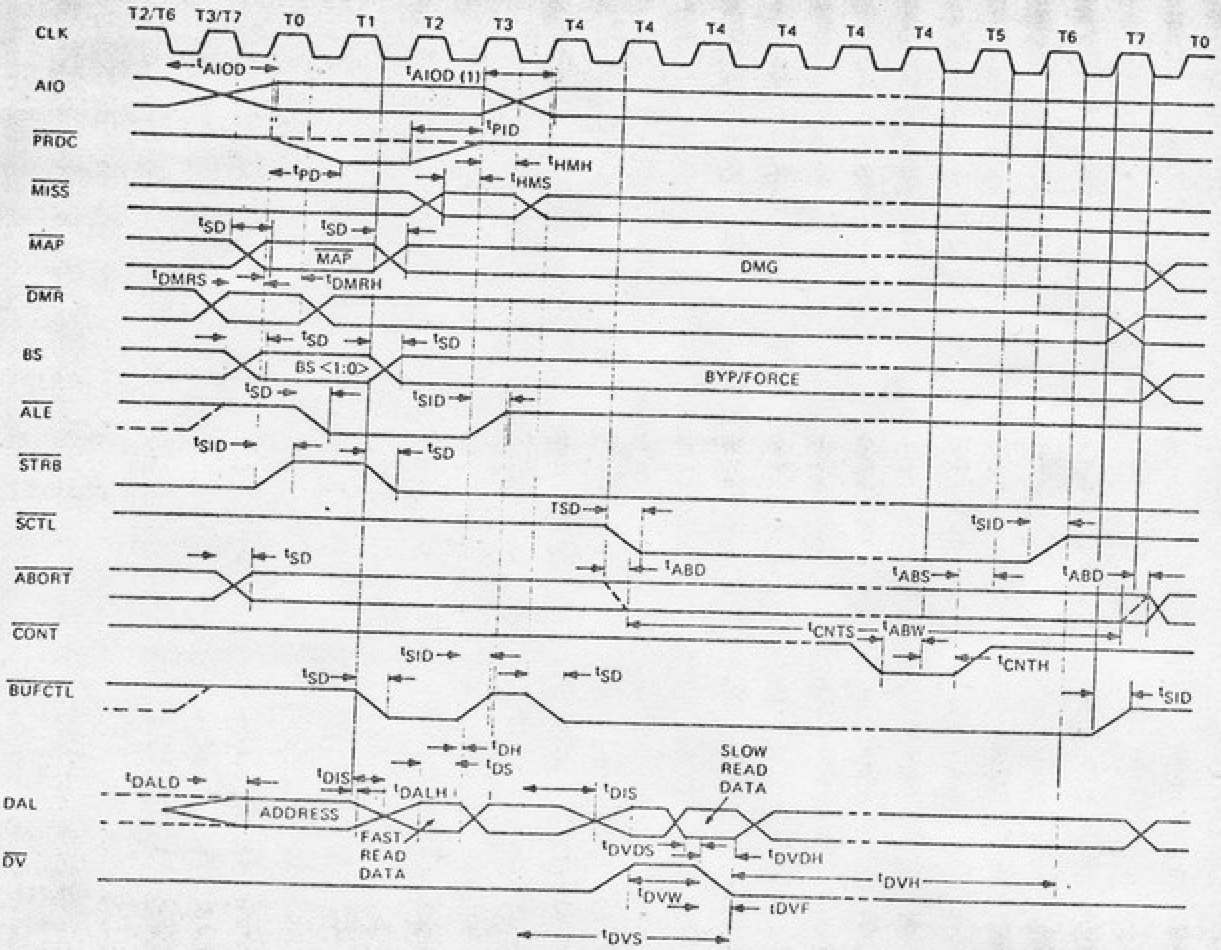


Figure B-6 Stretched Bus Read Timing

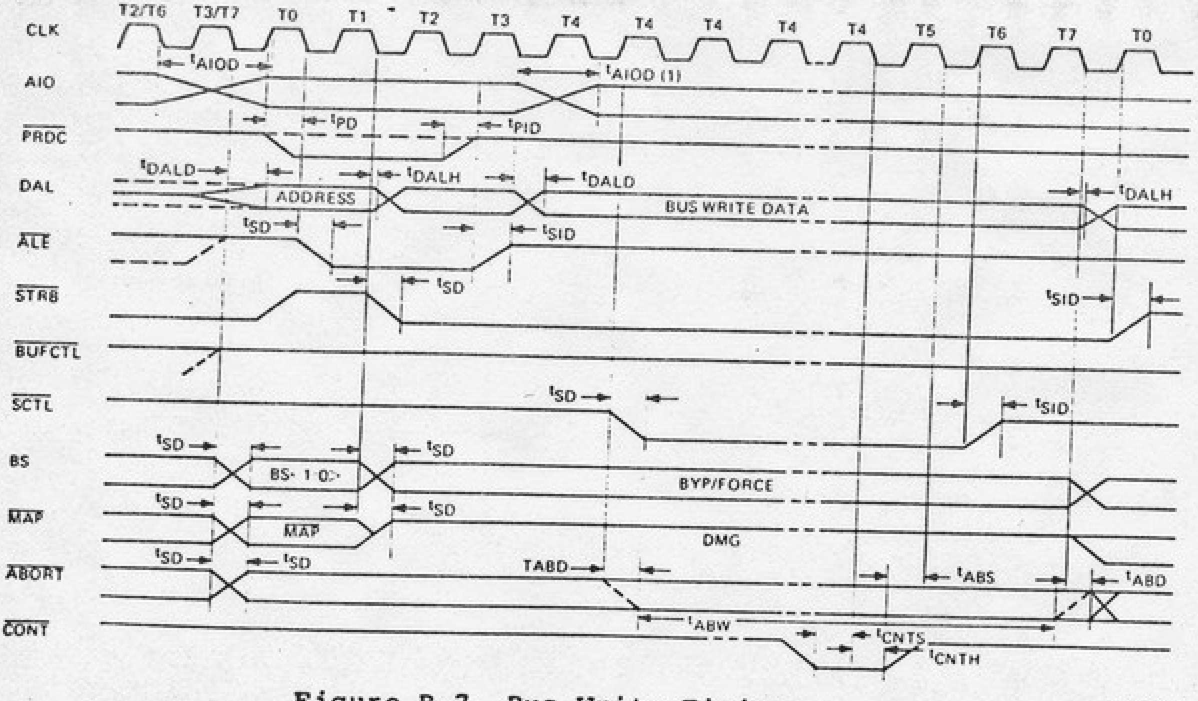


Figure B-7 Bus Write Timing

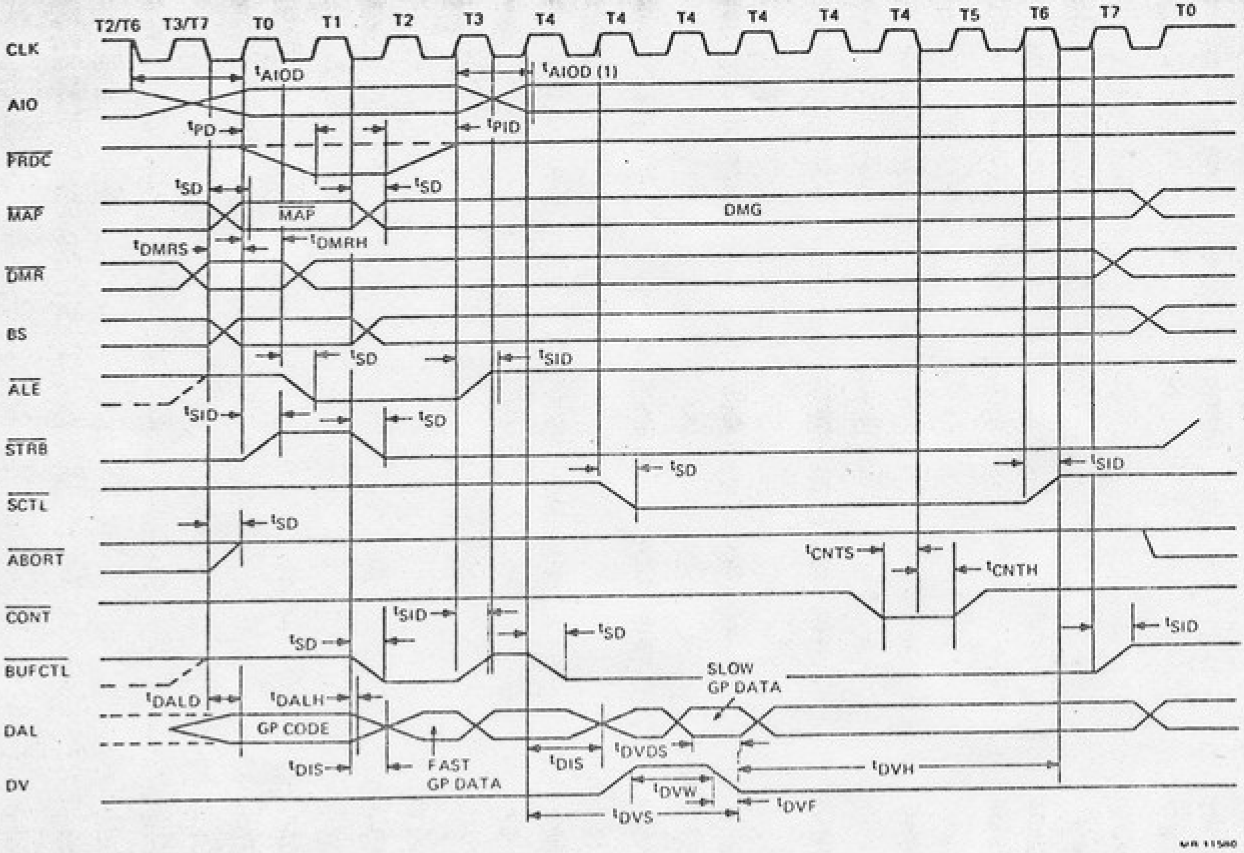


Figure B-8  General-Purpose Read Timing

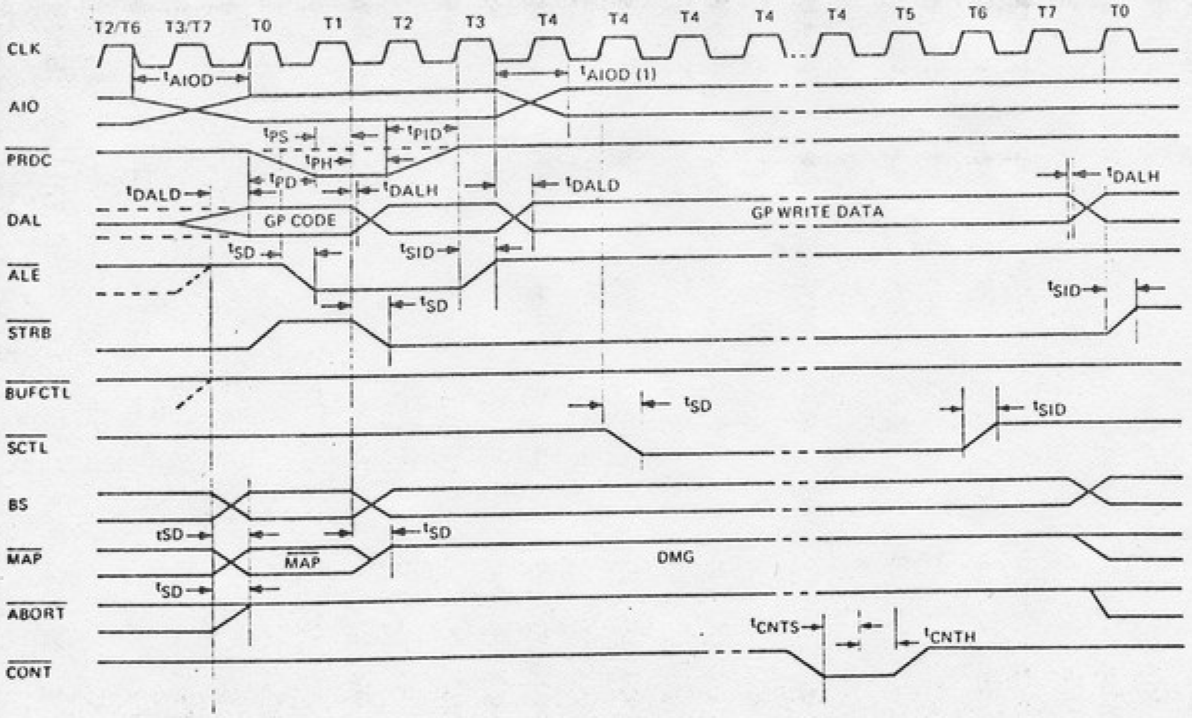


Figure B-9  General-Purpose Write Timing

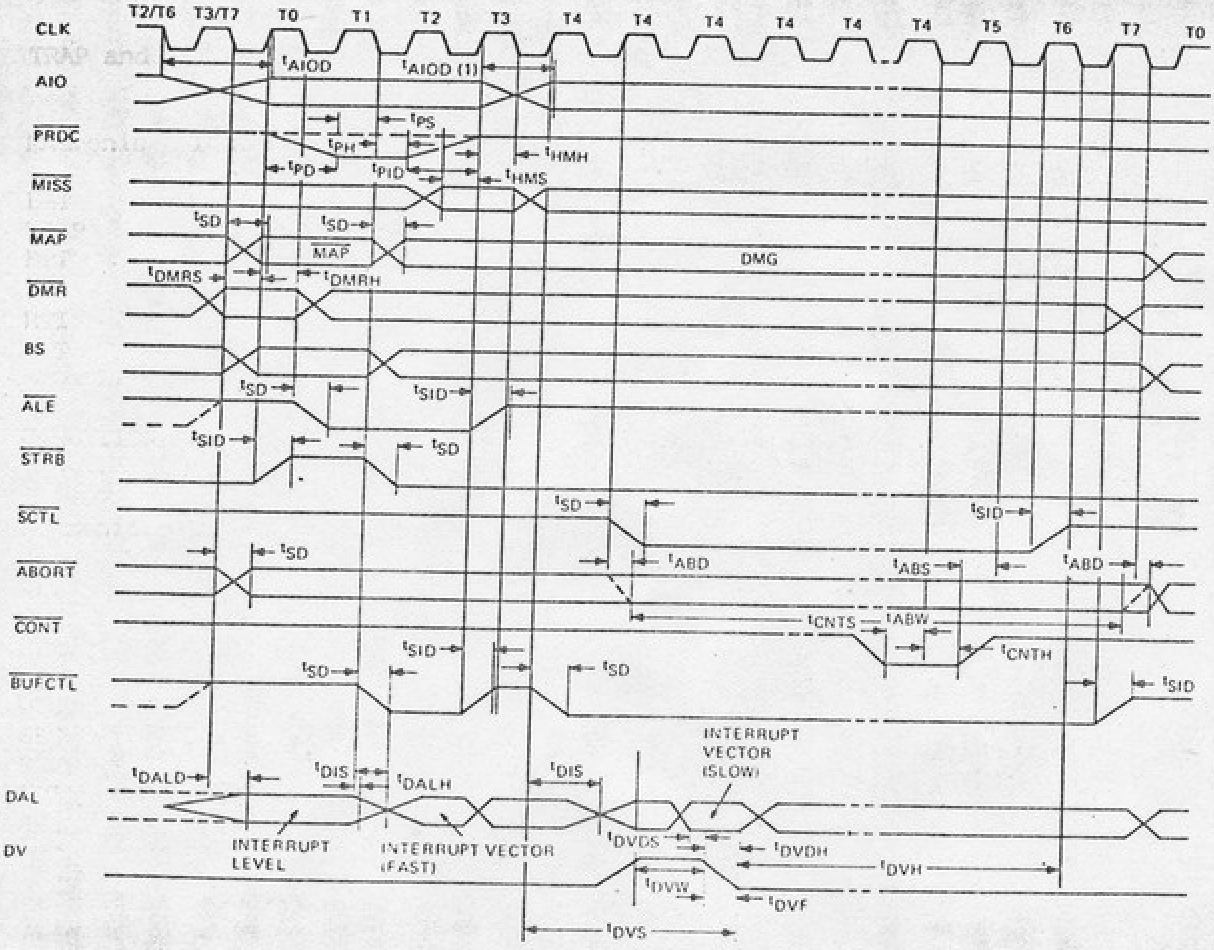


Figure B-10 Interrupt Acknowledge Timing

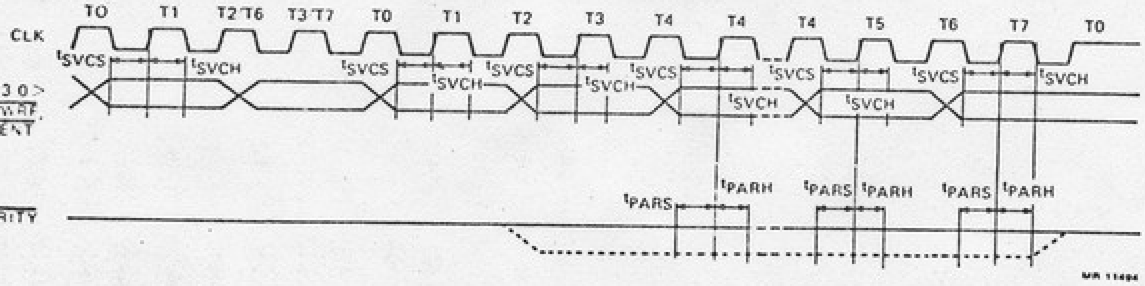


Figure B-11 Interrupt Timing

# APPENDIX D
# INSTRUCTION TIMING

The execution time for an instruction depends on: (1) the type of instruction executed, (2) the the mode of addressing used, and (3) the type of memory being referenced. In general, the total execution time is the sum of the base instruction fetch/execute time plus the operand(s) address calculation/fetch time.

The tables in this appendix can be used to calculate the length of an instruction in terms of microcycles (MC). In the first group of tables, the first column specifies the number of microcycles required to fetch/execute the base instruction. The R/W column specifies how many of these microcycles are read microcycles and how many are write microcycles (any remaining microcycles are NIO). If the instruction involves the calculation/fetch of one or more operands, a reference to a separate table (a source or destination table) is made in the last column(s). The source/destination tables reveal how many microcycles the source/destination calculation/fetch takes and also specifies how many of these are read or write microcycles (again, any remaining microcycles are NIO).

The numbers in the tables are based on the assumption that a memory read must last a minimum of four CLK periods, a memory write must last a minimum of eight CLK periods, and an NIO lasts four CLK periods (no DMA). Any wait states caused by slower memory or a DMA transfer must be added to the total instruction time. If wait states are required, the first wait state of a non-stretched read or NIO cycle will last four clock periods, and can continue in increments of two clock periods. Further wait states for stretched cycles occur in increments of two clock periods.

Floating-point instruction execution times are given as a range. The actual execution time will vary depending on the type of data being operated on.

Here are two examples of how to use the tables:

Example 1:

How long does a MOV R0,@#2044 instruction last?

Step 1: From the tables, the execution time for the MOV base instruction is found to be 1 microcycle (MC), or four CLK periods. This consists of one read and no write microcycles. Depending upon the type of memory in the system, the microcycle may be stretched. If so, the microcycle lasts at least eight CLK periods and may be stretched thereafter in increments of two CLK periods.

Step 2: To find the operand calculation/fetch time for the source operand (R0), refer to Table S1. From Table S1, it is seen that a mode 0 register 0 calculate/fetch takes 0 microcycles. Note that the operand is already available to the DCJ11 (in the register file).

Step 3: To find the operand calculation/fetch time for the destination operand (the contents of memory location 2044), refer to Table D3. From Table D3, it is seen that a mode 3 register 7 calculate/fetch takes 3 microcycles, one of which is a read microcycle and one of which is a write microcycle. Note that the remaining microcycle is an NIO microcycle. Once again, the type of memory in the system must be taken into account. If the read cycle is stretched, the stretched cycle lasts at least eight CLK periods and may be stretched thereafter in increments of two CLK periods. The write microcycle lasts at least eight CLK periods and may be stretched in increments of two CLK periods.

Step 4: For a determination of the minimum time required, total up the microcycles. In this example, It is 1 + 0 + 3, or 4 microcycles (which is 16 CLK periods if no microcycle stretching occurs).

Example 2:

The source and destination tables for floating point instructions show a negative number in the MC column for certain mode 2 register 7 operations. This example shows a timing calculation for one of these.

How long does an CLRD #2000 instruction last?

Step 1: The base instruction time for the CLRD instruction is 14 microcycles.

Step 2: From Table F2, the calculation/fetch time for the operand (a mode 2 register 7 reference) is shown as (-1). This means that one microcycle should be subtracted from the base instruction time. However, add one microcycle for the memory write operation. There are no memory read cycles to account for.

Step 3: Total up the microcycles: 14 - 1 + 1 = 14 microcycles minimum (assumes no cycle stretching).

SINGLE OPERAND

| Mnemonic | Instruction | TIMING<br>Execution<br>MC | <br><br>R/W | <br>Source<br>Table | <br>Dest<br>Table |
|---|---|---|---|---|---|
| General | | | | | |
| CLR(B) | Clear | 1 | 1/0 | — | D3 |
| COM(B) | Complement (1's) | 1 | 1/0 | — | D4 |
| INC(B) | Increment | 1 | 1/0 | — | D4 |
| DEC(B) | Decrement | 1 | 1/0 | — | D4 |
| NEG(B) | Negate (2's complement) | 1 | 1/0 | — | D4 |
| TST(B) | Test | 1 | 1/0 | — | D4 |
| Rotate and Shift | | | | | |
| ROR(B) | Rotate right | 1 | 1/0 | — | D4 |
| ROL(B) | Rotate left | 1 | 1/0 | — | D4 |
| ASR(B) | Arithmetic shift right | 1 | 1/0 | — | D4 |
| SWAB | Swap bytes | 1 | 1/0 | — | D4 |
| Multiple-Precision | | | | | |
| ADC(B) | Add carry | 1 | 1/0 | — | D4 |
| SBC(B) | Subtract carry | 1 | 1/0 | — | D4 |
| SXT | Sign extend | 1 | 1/0 | — | D3 |
| Multiprocessing | | | | | |
| TSTSET | Test and set<br>(low bit interlocked) | 5 | 1/1 | — | D4 |
| WRTLCK | Write interlocked | 4 | 1/1 | — | D4 |

DOUBLE OPERAND

| Mnemonic | Instruction | TIMING<br>Execution<br>MC | <br><br>R/W | <br>Source<br>Table | <br>Dest<br>Table |
|---|---|---|---|---|---|
| General | | | | | |
| MOV(B) | Move | 1 | 1/0 | S1 | D3 |
| CMP(B) | Compare | 1 | 1/0 | S1 | D2 |
| ADD | Add | 1 | 1/0 | S1 | D4 |
| SUB | Subtract | 1 | 1/0 | S1 | D4 |
| Logical | | | | | |
| BIT(B) | Bit test (AND) | 1 | 1/0 | S1 | D2 |
| BIC(B) | Bit clear | 1 | 1/0 | S1 | D4 |
| BIS(B) | Bit set (OR) | 1 | 1/0 | S1 | D4 |

Register

| | | | | | | |
|---|---|---|---|---|---|---|
| MUL | Multiply | 22 | 1/0 | — | D1 | (Notes 5,11) |
| DIV | Divide | 34 | 1/0 | — | D1 | (Notes 6,7,12) |
| ASH | Shift automatically | 4 | 1/0 | — | D1 | |
| ASHC | Arith shift combined | 5 | 1/0 | — | D1 | (Note 13) |
| XOR | Exclusive (OR) | 1 | 1/0 | — | D4 | |

BRANCH

| | | TIMING | | | |
|---|---|---|---|---|---|
| | | Branch | | Branch | |
| | | Not Taken | | Taken | |
| Mnemonic | Instruction | MC | R/W | MC | R/W |
| Branches | | | | | |
| BR | Branch (unconditional) | 2 | 1/0 | 4 | 2/0 |
| BNE | Br if not equal (to 0) | 2 | 1/0 | 4 | 2/0 |
| BEQ | Br if equal (to 0) | 2 | 1/0 | 4 | 2/0 |
| BPL | Br if plus | 2 | 1/0 | 4 | 2/0 |
| BMI | Br if minus | 2 | 1/0 | 4 | 2/0 |
| BVC | Br if overflow is clear | 2 | 1/0 | 4 | 2/0 |
| BVS | Br if overflow is set | 2 | 1/0 | 4 | 2/0 |
| BCC | Br if carry is clear | 2 | 1/0 | 4 | 2/0 |
| BCS | Br if carry is set | 2 | 1/0 | 4 | 2/0 |
| Signed Conditional Branches | | | | | |
| BGE | Br if greater or equal (to 0) | 2 | 1/0 | 4 | 2/0 |
| BLT | Br if less than (0) | 2 | 1/0 | 4 | 2/0 |
| BGT | Br if greater than (0) | 2 | 1/0 | 4 | 2/0 |
| BLE | Br if less or equal (to 0) | 2 | 1/0 | 4 | 2/0 |

| | | Branch | | Branch | |
|---|---|---|---|---|---|
| | | Not Taken | | Taken | |
| Mnemonic | Instruction | MC | R/W | MC | R/W |
| Unsigned Conditional Branches | | | | | |
| BHI | Branch if higher | 2 | 1/0 | 4 | 2/0 |
| BLOS | Branch if lower or same | 2 | 1/0 | 4 | 2/0 |
| BHIS | Branch if higher or same | 2 | 1/0 | 4 | 2/0 |
| BLO | Branch if lower | 2 | 1/0 | 4 | 2/0 |
| SOB | Subtract 1 and branch (if ≠ 0) | 3 | 1/0 | 5 | 2/0 |

JUMP and SUBROUTINE

| | | TIMING | | | |
|---|---|---|---|---|---|
| | | Execution | | | |
| Mnemonic | Instruction | MC | R/W | DST Table | |
| JMP | Jump | — | — | D5 | |
| JSR | Jump to subroutine | — | — | D6 | (Note 4) |
| RTS | Return from subroutine | 5 | 3/0 | — | (Note 14) |
| MARK | Stack cleanup | 10 | 3/0 | | |

## TRAP and INTERRUPT

| | | TIMING | |
|---|---|---|---|
| | | Execution | |
| Mnemonic | Instruction | MC | R/W |
| EMT | Emulator trap | 20 | 4/2 |
| TRAP | Trap | 20 | 4/2 |
| BPT | Breakpoint trap | 20 | 4/2 |
| IOT | Input/output trap | 20 | 4/2 |
| RTI | Return from interrupt | 9 | 4/0 |
| RTT | Return from interrupt | 9 | 4/0 |

## CONDITION CODE OPERATORS

| | | TIMING | |
|---|---|---|---|
| | | Execution | |
| Mnemonic | Instruction | MC | R/W |
| CLC | Clear C | 3 | 1/0 |
| CLV | Clear V | 3 | 1/0 |
| CLZ | Clear Z | 3 | 1/0 |
| CLN | Clear N | 3 | 1/0 |
| CCC | Clear all CC bits | 3 | 1/0 |
| SEC | Set C | 3 | 1/0 |
| SEV | Set V | 3 | 1/0 |
| SEZ | Set Z | 3 | 1/0 |
| SEN | Set N | 3 | 1/0 |
| SCC | Set all CC bits | 3 | 1/0 |

## MISCELLANEOUS

| | | TIMING | | |
|---|---|---|---|---|
| | | Execution | | Dest |
| Mnemonic | Instruction | MC | R/W | Table |
| HALT | Halt | - | | — |
| WAIT | Wait for interrupt | - | | — |
| RESET | Reset external bus | - | | — |
| NOP | (No operation) | 3 | 1/0 | — |
| SPL | Set priority level to N | 7 | 1/0 | — |
| MFPI | Move from previous instr space | 5 | 1/1 | D1 |
| MTPI | Move to previous instr space | 3 | 2/0 | D3 |
| MFPD | Move from previous data space | 5 | 1/1 | D1 |
| MTPD | Move to previous data space | 3 | 2/0 | D3 |
| MTPS | Move byte to PSW PS ∢ (svc) | 8 | 1/0 | D1 |
| MFPS | Move byte from PSW (dst) ∢ PS <7:0> | 1 | 1/0 | D3 |
| MFPT | Move from processor (R0<7:0>∢proc code | 2 | 1/0 | — |
| CSM | Call to supervisor mode | 28 | 3/3 | D1 |

FLOATING POINT TIMING

| | | Execution (MC) | | | Non Mode 0 |
|---|---|---|---|---|---|
| Mnemonic | Instruction | Min | Typ | Max | Table |
| ABSD | Make Absolute | 23 | | 24 | F3 |
| ABSF | Make Absolute | 19 | | 20 | F3 |
| ADDD | Add | 41 | 48 | 119 | F1 |
| ADDF | Add | 31 | 35 | 102 | F1 |
| CFCC | Copy Floating Condition Codes | 5 | | 5 | — |
| CLRD | Clear | 14 | | 14 | F2 |
| CLRF | Clear | 12 | | 12 | F2 |
| CMPD | Compare | 24 | | 25 | F1 |
| CMPF | Compare | 18 | | 19 | F1 |
| DIVD | Divide | 160 | | 167 | F1 |
| DIVF | Divide | 59 | | 63 | F1 |
| LDCDF | Ld & C from D to F | 24 | | 26 | F1 |
| LDCFD | Ld & C from F to D | 20 | | 21 | F1 |
| LDCID | Ld & C Integer to D | 31 | | 42 | F4 |
| LDCIF | Ld & C Integer to F | 26 | | 36 | F4 |
| LDCLD | Ld & C Long Integer to D | 31 | | 52 | F4 |
| LDCLF | Ld & C Long Integer to F | 26 | | 44 | F4 |
| LDD | Load | 16 | | 17 | F1 |
| LDEXP | Load Exponent | 17 | | 18 | F4 |
| LDF | Load | 12 | | 13 | F1 |
| LDFPS | Load FPP Program Status | 6 | | 6 | F4 |
| MODD | Multiply and Separate | 202 | 217 | 268 | F1 |
| MODF | Integer and Fraction | 82 | 94 | 115 | F1 |
| MULD | Multiply | 165 | | 173 | F1 |
| MULF | Multiply | 56 | | 61 | F1 |
| NEGD | Negate | 22 | | 23 | F3 |
| NEGE | Negate | 18 | | 19 | F3 |
| SETD | Set Floating Double Mode | 6 | | 6 | — |
| SETF | Set Floating Mode | 6 | | 6 | — |
| SETI | Set Integer Mode | 6 | | 6 | — |
| SETL | Set Long Integer Mode | 6 | | 6 | — |
| STCDF | St & C from D to F | 17 | | 20 | F2 |
| STCDI | St & C from D to Integer | 26 | | 38 | F5 |
| STCDL | St & C from D to Long Integer | 26 | | 54 | F5 |
| STCFD | St & C from F to D | 19 | | 20 | F2 |
| STCFI | St & C from F to Integer | 23 | | 35 | F5 |
| STCFL | St & C from F to Long Integer | 23 | | 51 | F5 |
| STD | Store | 12 | | 12 | F2 |
| STECP | Store Exponent | 16 | | 16 | F5 |
| STF | Store | 8 | | 8 | F2 |
| STFPD | Store FPP Program Status | 9 | | 9 | F5 |
| STST | Store FPP Status | 7 | | 7 | F5 |
| SUBD | Subtract | 47 | 55 | 122 | F1 |
| SUBF | Subtract | 37 | 41 | 104 | F1 |
| TSTD | Test | 11 | | 12 | F1 |
| TSTF | Test | 9 | | 10 | F1 |

SOURCE AND DESTINATION TABLES:

Table S1 Source Address Time: All Double Operand

| Source Mode | Source Register | Microcode Cycles | Read Memory Cycles |
|---|---|---|---|
| 0 | 0-7 | 0 | 0 |
| 1 | 0-7 | 2 | 1 |
| 2 | 0-6 | 2 | 1 |
| 2 | 7 | 1 | 1 |
| 3 | 0-6 | 4 | 2 |
| 3 | 7 | 3 | 2 |
| 4 | 0-6 | 3 | 1 |
| 4 | 7 | 6 | 2 (Note 1) |
| 5 | 0-6 | 5 | 2 |
| 5 | 7 | 8 | 3 (Note 1) |
| 6 | 0-7 | 4 | 2 |
| 7 | 0-7 | 6 | 3 |

Table D1 Destination Address Time: Read Only Single Operand

| Destination Mode | Destination Register | Microcode Cycles | Read Memory Cycles |
|---|---|---|---|
| 0 | 0-7 | 0 | 0 |
| 1 | 0-7 | 2 | 1 |
| 2 | 0-6 | 2 | 1 |
| 2 | 7 | 1 | 1 |
| 3 | 0-6 | 4 | 2 |
| 3 | 7 | 3 | 2 |
| 4 | 0-6 | 3 | 1 |
| 4 | 7 | 7 | 2 (Note 2) |
| 5 | 0-6 | 5 | 2 |
| 5 | 7 | 9 | 3 (Note 3) |
| 6 | 0-7 | 4 | 2 |
| 7 | 0-7 | 6 | 3 |

Table D2 Destination Address Time: Read Only Double Operand

| Destination Mode | Destination Register | Microcode Cycles | Read Memory Cycles |
|---|---|---|---|
| 0 | 0-7 | 0 | 0 |
| 1 | 0-7 | 3 | 1 |
| 2 | 0-6 | 3 | 1 |
| 2 | 7 | 2 | 1 |
| 3 | 0-6 | 5 | 2 |
| 3 | 7 | 4 | 2 |
| 4 | 0-6 | 4 | 1 |
| 4 | 7 | 8 | 2 (Note 2) |
| 5 | 0-6 | 6 | 2 |
| 5 | 7 | 10 | 3 (Note 3) |
| 6 | 0-7 | 5 | 2 |
| 7 | 0-7 | 7 | 3 |

Table D3 Destination Address Time: Write Only

| Destination Mode | Destination Register | Microcode Cycles | Memory Cycles | |
|---|---|---|---|---|
| | | | Read | Write |
| 0 | 0-6 | 0 | 0 | 0 |
| 0 | 7 | 5 | 1 | 0 |
| 1 | 0-6 | 2 | 0 | 1 |
| 1 | 7 | 6 | 1 | 1 |
| 2 | 0-6 | 2 | 0 | 1 |
| 2 | 7 | 6 | 1 | 1 |
| 3 | 0-6 | 4 | 1 | 1 |
| 3 | 7 | 3 | 1 | 1 |
| 4 | 0-6 | 3 | 0 | 1 |
| 4 | 7 | 7 | 1 | 1 |
| 5 | 0-6 | 5 | 1 | 1 |
| 5 | 7 | 9 | 2 | 1 |
| 6 | 0-7 | 4 | 1 | 1 |
| 7 | 0-7 | 6 | 2 | 1 |

Table D4 Destination Address Time: Read Modify Write

| Destination Mode | Destination Register | Microcode Cycles | Memory Cycles | | |
|---|---|---|---|---|---|
| | | | Read | Write | |
| 0 | 0-6 | 0 | 0 | 0 | |
| 0 | 7 | 5 | 1 | 0 | |
| 1 | 0-6 | 3 | 1 | 1 | |
| 1 | 7 | 7 | 2 | 1 | |
| 2 | 0-6 | 3 | 1 | 1 | |
| 2 | 7 | 7 | 2 | 1 | |
| 3 | 0-6 | 5 | 2 | 1 | |
| 3 | 7 | 4 | 2 | 1 | |
| 4 | 0-6 | 4 | 1 | 1 | |
| 4 | 7 | 8 | 2 | 1 | (Note 2) |
| 5 | 0-6 | 6 | 2 | 1 | |
| 5 | 7 | 10 | 3 | 1 | (Note 3) |
| 6 | 0-7 | 5 | 2 | 1 | |
| 7 | 0-7 | 7 | 3 | 1 | |

Table D5 Destination Address Time: JMP

| Destination Mode | Destination Register | Microcode Cycles | Memory Cycles | |
|---|---|---|---|---|
| | | | Read | Write |
| 1 | 0-7 | 4 | 2 | 0 |
| 2 | 0-7 | 6 | 2 | 0 |
| 3 | 0-7 | 5 | 3 | 0 |
| 4 | 0-7 | 5 | 2 | 0 |
| 5 | 0-7 | 6 | 3 | 0 |
| 6 | 0-6 | 6 | 3 | 0 |
| 6 | 7 | 5 | 3 | 0 |
| 7 | 0-7 | 7 | 4 | 0 |

Table D6   Destination Address Time:  JSR

| Destination Mode | Destination Register | Microcode Cycles | Memory Cycles Read | Memory Cycles Write |
|---|---|---|---|---|
| 1 | 0-7 | 9 | 2 | 1 |
| 2 | 0-7 | 10 | 2 | 1 |
| 3 | 0-6 | 10 | 3 | 1 |
| 3 | 7 | 9 | 3 | 1 |
| 4 | 0-7 | 10 | 2 | 1 |
| 5 | 0-7 | 11 | 3 | 1 |
| 6 | 0-6 | 10 | 3 | 1 |
| 6 | 7 | 9 | 3 | 1 |
| 7 | 0-7 | 12 | 4 | 1 |

Table F1   Floating Source Modes 1-7

Single Precision

| Mode | Register | Microcode Cycles | Memory Read | Memory Write |
|---|---|---|---|---|
| 1 | 0-7 | 3 | 2 | 0 |
| 2 | 0-6 | 3 | 2 | 0 |
| 2 | 7 | 1 | 1 | 0 |
| 3 | 0-6 | 4 | 3 | 0 |
| 3 | 7 | 3 | 3 | 0 |
| 4 | 0-7 | 4 | 2 | 0 |
| 5 | 0-7 | 5 | 3 | 0 |
| 6 | 0-7 | 4 | 3 | 0 |
| 7 | 0-7 | 6 | 4 | 0 |

Double Precision

| Mode | Register | Microcode Cycles | Memory Read | Memory Write |
|---|---|---|---|---|
| 1 | 0-7 | 5 | 4 | 0 |
| 2 | 0-6 | 5 | 4 | 0 |
| 2 | 7 | 0 (Note 15) | 1 | 0 |
| 3 | 0-6 | 6 | 5 | 0 |
| 3 | 7 | 5 | 5 | 0 |
| 4 | 0-7 | 6 | 4 | 0 |
| 5 | 0-7 | 7 | 5 | 0 |
| 6 | 0-7 | 6 | 5 | 0 |
| 7 | 0-7 | 8 | 6 | 0 |

## Table F2 Floating Destination Modes 1-7

### Single Precision

| Mode | Register | Microcode Cycles | Memory Read | Memory Write |
|---|---|---|---|---|
| 1 | 0-7 | 3 | 0 | 2 |
| 2 | 0-6 | 3 | 0 | 2 |
| 2 | 7 | 1 | 0 | 1 |
| 3 | 0-6 | 4 | 1 | 2 |
| 3 | 7 | 3 | 1 | 2 |
| 4 | 0-7 | 4 | 0 | 2 |
| 5 | 0-7 | 5 | 1 | 2 |
| 6 | 0-7 | 4 | 1 | 2 |
| 7 | 0-7 | 6 | 2 | 2 |

### Double Precision

| Mode | Register | Microcode Cycles | Memory Read | Memory Write |
|---|---|---|---|---|
| 1 | 0-7 | 5 | 0 | 4 |
| 2 | 0-6 | 5 | 0 | 4 |
| 2 | 7 | (-1) (Note 15) | 0 | 1 |
| 3 | 0-6 | 6 | 1 | 4 |
| 3 | 7 | 5 | 1 | 4 |
| 4 | 0-7 | 6 | 0 | 4 |
| 5 | 0-7 | 7 | 1 | 4 |
| 6 | 0-7 | 6 | 1 | 4 |
| 7 | 0-7 | 8 | 2 | 4 |

## Table F3 Floating Read Modify Write Modes 1-7

### Single Precision

| Mode | Register | Microcode Cycles | Memory Read | Memory Write |
|---|---|---|---|---|
| 1 | 0-7 | 5 | 2 | 2 |
| 2 | 0-6 | 5 | 2 | 2 |
| 2 | 7 | 1 (Note 15) | 1 | 1 |
| 3 | 0-6 | 6 | 3 | 2 |
| 3 | 7 | 5 | 3 | 2 |
| 4 | 0-7 | 6 | 2 | 2 |
| 5 | 0-7 | 7 | 3 | 2 |
| 6 | 0-7 | 6 | 3 | 2 |
| 7 | 0-7 | 8 | 4 | 2 |

## Table F3 Floating Read Modify Write Modes 1-7

### Double Precision

| Mode | Register | Microcode Cycles | Memory Read | Memory Write |
|---|---|---|---|---|
| 1 | 0-7 | 9 | 4 | 4 |
| 2 | 0-6 | 9 | 4 | 4 |
| 2 | 7 | (-2) (Note 15) | 1 | 1 |
| 3 | 0-6 | 10 | 5 | 4 |
| 3 | 7 | 9 | 5 | 4 |
| 4 | 0-7 | 10 | 4 | 4 |
| 5 | 0-7 | 11 | 5 | 4 |
| 6 | 0-7 | 10 | 5 | 4 |
| 7 | 0-7 | 12 | 6 | 4 |

## Table F4 Integer Source Modes 1-7

### Integer

| Mode | Register | Microcode Cycles | Memory Read | Memory Write |
|---|---|---|---|---|
| 1 | 0-7 | 2 | 1 | 0 |
| 2 | 0-6 | 2 | 1 | 0 |
| 2 | 7 | 0 (Note 15) | 1 | 0 |
| 3 | 0-6 | 3 | 2 | 0 |
| 3 | 7 | 2 | 2 | 0 |
| 4 | 0-7 | 3 | 1 | 0 |
| 5 | 0-7 | 4 | 2 | 0 |
| 6 | 0-7 | 3 | 2 | 0 |
| 7 | 0-7 | 5 | 3 | 0 |

### Long Integer

| Mode | Register | Microcode Cycles | Memory Read | Memory Write |
|---|---|---|---|---|
| 1 | 0-7 | 4 | 2 | 0 |
| 2 | 0-6 | 4 | 2 | 0 |
| 2 | 7 | 0 (Note 15) | 1 | 0 |
| 3 | 0-6 | 5 | 3 | 0 |
| 3 | 7 | 4 | 3 | 0 |
| 4 | 0-7 | 5 | 2 | 0 |
| 5 | 0-7 | 6 | 3 | 0 |
| 6 | 0-7 | 5 | 3 | 0 |
| 7 | 0-7 | 7 | 4 | 0 |

Table F5 Integer Destination Modes 1-7

Integer

| Mode | Register | Microcode Cycles | Memory Read | Memory Write |
|---|---|---|---|---|
| 1 | 0-7 | 2 | 0 | 1 |
| 2 | 0-6 | 2 | 0 | 1 |
| 2 | 7 | 2 | 0 | 1 |
| 3 | 0-6 | 3 | 1 | 1 |
| 3 | 7 | 2 | 1 | 1 |
| 4 | 0-7 | 3 | 0 | 1 |
| 5 | 0-7 | 4 | 1 | 1 |
| 6 | 0-7 | 3 | 1 | 1 |
| 7 | 0-7 | 5 | 2 | 1 |

Long Integer

| Mode | Register | Microcode Cycles | Memory Read | Memory Write |
|---|---|---|---|---|
| 1 | 0-7 | 4 | 0 | 2 |
| 2 | 0-6 | 4 | 0 | 2 |
| 2 | 7 | 2 | 0 | 1 |
| 3 | 0-6 | 5 | 1 | 2 |
| 3 | 7 | 4 | 1 | 2 |
| 4 | 0-7 | 5 | 0 | 2 |
| 5 | 0-7 | 6 | 1 | 2 |
| 6 | 0-7 | 5 | 1 | 2 |
| 7 | 0-7 | 7 | 2 | 2 |

## NOTES

1. Subtract 2 microcycles (MC) and one read if both source and destination modes autodecrement PC, or if WRITE-ONLY or READ-MODIFY-WRITE mode 07 or 17 is used.

2. READ-ONLY and READ-MODIFY-WRITE destination mode 47 references actually perform 3 READ operations. For bookkeeping purposes, one of the READs is accounted for in the EXECUTE, FETCH TIMING.

3. READ-ONLY and READ-MODIFY-WRITE destination mode 57 references actually perform 4 READ operations. For bookkeeping purposes, one of the READs is accounted for in the EXECUTE, FETCHING TIMING.

4. Subtract 1 MC if the link register is PC.

5. Add 1 MC if the source operand is negative.

6. Subtract 1 MC if the source mode is not zero.

7. Add 1 MC if the quotient is even.
   Add 2 MC if overflow occurs.
   Add 5 MC and 1 read if the PC is used as a destination register, but only if source mode 47 or 57 is not used.

8. Add 1 MC per shift.

9. Add 1 MC if source operand <15:6> is not zero.

10. Subtract 1 MC if one shift only.

11. Add 4 MC and 1 read if the PC is used as a destination register, but only if source mode 47 or 57 is not used.

12. Divide by zero executes in 5 MC (see note 6).

13. Timing for no shift. Add 1 MC if a left shift. (Notes 8, 9, 11 apply.) Add 2 MC for a right shift. (Notes 8, 10, 11 apply.)

14. Add one MC if a register other than R7 is used.

15. Mode 27 references only access single word operands. The execution time listed has been compensated in order to accurately compute the total execution time.

57

Part No. : 32 958 044
Related doc: 2

# Z8530 SCC Serial Communications Controller

Zilog

## Product Specification

September 1983

### Features

- Two independent, 0 to 1M bit/second, full-duplex channels, each with a separate crystal oscillator, baud rate generator, and Digital Phase-Locked Loop for clock recovery.
- Multi-protocol operation under program control; programmable for NRZ, NRZI, or FM data encoding.
- Asynchronous mode with five to eight bits and one, one and one-half, or two stop bits per character; programmable clock factor; break detection and generation; parity, overrun, and framing error detection.
- Synchronous mode with internal or external character synchronization on one or two synchronous characters and CRC generation and checking with CRC-16 or CRC-CCITT preset to either 1s or 0s.
- SDLC/HDLC mode with comprehensive frame-level control, automatic zero insertion and deletion, I-field residue handling, abort generation and detection, CRC generation and checking, and SDLC Loop mode operation.
- Local Loopback and Auto Echo modes.

### General Description

The Z8530 SCC Serial Communications Controller is a dual-channel, multi-protocol data communications peripheral designed for use with conventional non-multiplexed buses. The SCC functions as a serial-to-parallel, parallel-to-serial converter/controller. The SCC can be software-configured to satisfy a wide variety of serial communications applications. The device contains a variety of new, sophisticated internal functions including on-chip baud rate generators, Digital Phase-Locked Loops, and crystal oscillators that dramatically reduce the need for external logic.

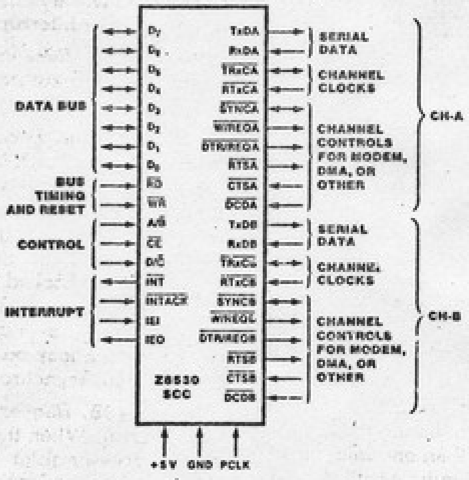


**Figure 1. Pin Functions**

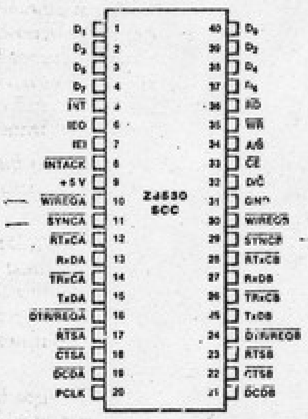


**Figure 2. Pin Assignments**

## General Description (Continued)

The SCC handles asynchronous formats, Synchronous byte-oriented protocols such as IBM Bisync, and Synchronous bit-oriented protocols such as HDLC and IBM SDLC. This versatile device supports virtually any serial data transfer application (cassette, diskette, tape drives, etc.).

The device can generate and check CRC codes in any Synchronous mode and can be programmed to check data integrity in various modes. The SCC also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

The Z-Bus daisy-chain interrupt hierarchy is also supported—as is standard for Zilog peripheral components.

The Z8530 SCC is packaged in a 40-pin ceramic DIP and uses a single +5 V power supply.

## Pin Description

The following section describes the pin functions of the SCC. Figures 1 and 2 detail the respective pin functions and pin assignments.

**A/$\overline{\mathrm{B}}$.** *Channel A/Channel B Select* (input). This signal selects the channel in which the read or write operation occurs.

**$\overline{\mathrm{CE}}$.** *Chip Enable* (input, active Low). This signal selects the SCC for a read or write operation.

**$\overline{\mathrm{CTSA}}$, $\overline{\mathrm{CTSB}}$.** *Clear To Send* (inputs, active Low). If these pins are programmed as Auto Enables, a Low on the inputs enables the respective transmitters. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The SCC detects pulses on these inputs and can interrupt the CPU on both logic level transitions.

**D/$\overline{\mathrm{C}}$.** *Data/Control Select* (input). This signal defines the type of information transferred to or from the SCC. A High means data is transferred; a Low indicates a command.

**$\overline{\mathrm{DCDA}}$, $\overline{\mathrm{DCDB}}$.** *Data Carrier Detect* (inputs, active Low). These pins function as receiver enables if they are programmed for Auto Enables; otherwise they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accomodate slow rise-time signals. The SCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.

**$D_0$-$D_7$.** *Data Bus* (bidirectional, 3-state). These lines carry data and commands to and from the SCC.

**$\overline{\mathrm{DTR}}/\overline{\mathrm{REQA}}$, $\overline{\mathrm{DTR}}/\overline{\mathrm{REQB}}$.** *Data Terminal Ready/Request* (outputs, active Low). These outputs follow the state programmed into the DTR bit. They can also be used as general-purpose outputs or as Request lines for a DMA controller.

**IEI.** *Interrupt Enable In* (input, active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A High IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.

**IEO.** *Interrupt Enable Out* (output, active High). IEO is High only if IEI is High and the CPU is not servicing an SCC interrupt or the SCC is not requesting an interrupt (Interrupt Acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.

**$\overline{\mathrm{INT}}$.** *Interrupt Request* (output, open-drain, active Low). This signal is activated when the SCC requests an interrupt.

**$\overline{\mathrm{INTACK}}$.** *Interrupt Acknowledge* (input, active Low). This signal indicates an active Interrupt Acknowledge cycle. During this cycle, the SCC interrupt daisy chain settles. When $\overline{\mathrm{RD}}$ becomes active, the SCC places an interrupt vector on the data bus (if IEI is High). $\overline{\mathrm{INTACK}}$ is latched by the rising edge of PCLK.

**PCLK.** *Clock* (input). This is the master SCC clock used to synchronize internal signals PCLK is a TTL level signal.

**$\overline{\mathrm{RD}}$.** *Read* (input, active Low). This signal indicates a read operation and when the SCC is selected, enables the SCC's bus drivers. During the Interrupt Acknowledge cycle, this signal gates the interrupt vector onto the bus if the SCC is the highest priority device requesting an interrupt.

**RxDA, RxDB.** *Receive Data* (inputs, active High). These input signals receive serial data at standard TTL levels.

**$\overline{\mathrm{RTxCA}}$, $\overline{\mathrm{RTxCB}}$.** *Receive/Transmit Clocks* (inputs, active Low). These pins can be programmed in several different modes of operation. In each channel, $\overline{\mathrm{RTxC}}$ may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the Digital Phase-Locked Loop. These pins can also be programmed for use with the respective $\overline{\mathrm{SYNC}}$ pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in Asynchronous modes.

**$\overline{\mathrm{RTSA}}$, $\overline{\mathrm{RTSB}}$.** *Request To Send* (outputs, active Low). When the Request To Send (RTS) bit in Write Register 5 (Figure 11) is set, the $\overline{\mathrm{RTS}}$ signal goes Low. When the RTS bit is reset in the Asynchronous mode and Auto

**Pin Description** (Continued)

Enable is on, the signal goes High after the transmitter is empty. In Synchronous mode or in Asynchronous mode with Auto Enable off, the RTS pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.

**$\overline{\mathrm{SYNCA}}$, $\overline{\mathrm{SYNCB}}$.** *Synchronization* (inputs or outputs, active Low). These pins can act either as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to $\overline{\mathrm{CTS}}$ and $\overline{\mathrm{DCD}}$. In this mode, transitions on these lines affect the state of the Synchronous/Hunt status bits in Read Register 0 (Figure 10) but have no other function.

In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, $\overline{\mathrm{SYNC}}$ must be driven Low two receive clock cycles after the last bit in the synchronous character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of $\overline{\mathrm{SYNC}}$.

In the Internal Synchronization mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which synchronous characters are recognized. The synchronous condition is not latched, so these outputs are active each time a synchronization pattern is recognized (regardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.

**TxDA, TxDB.** *Transmit Data* (outputs, active High). These output signals transmit serial data at standard TTL levels.

**$\overline{\mathrm{TRxCA}}$, $\overline{\mathrm{TRxCB}}$.** *Transmit/Receive Clocks* (inputs or outputs, active Low). These pins can be programmed in several different modes of operation. $\overline{\mathrm{TRxC}}$ may supply the receive clock or the transmit clock in the input mode or supply the output of the Digital Phase-Locked Loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.

**$\overline{\mathrm{WR}}$.** *Write* (input, active Low). When the SCC is selected, this signal indicates a write operation. The coincidence of $\overline{\mathrm{RD}}$ and $\overline{\mathrm{WR}}$ is interpreted as a reset.

**$\mathrm{W}/\overline{\mathrm{REQA}}$, $\mathrm{W}/\overline{\mathrm{REQB}}$.** *Wait/Request* (outputs, open-drain when programmed for a Wait function, driven High or Low when programmed for a Request function). These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the SCC data rate. The reset state is Wait.

**Functional Description**

The functional capabilities of the SCC can be described from two different points of view: as a data communications device, it transmits and receives data in a wide variety of data communications protocols; as a microprocessor peripheral, the SCC offers valuable features such as vectored interrupts, polling, and simple handshake capability.

**Data Communications Capabilities.** The SCC provides two independent full-duplex channels programmable for use in any common Asynchronous or Synchronous data-communication protocol. Figure 3 and the following description briefly detail these protocols.

*Asynchronous Modes.* Transmission and reception can be accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one-and-a-half, or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection

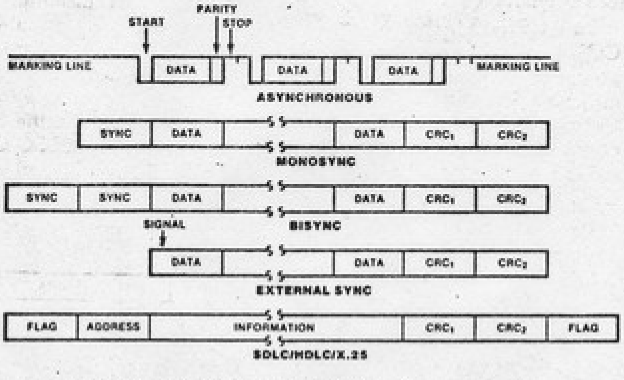


Figure 3. Some SCC Protocols

**Functional Description (Continued)**

mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB in Figure 1). If the Low does not persist (as in the case of a transient), the character assembly process does not start.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing or error conditions using dedicated routines. Furthermore, a built-in checking process avoids the interpretation of a framing error as a new start bit: a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit begins.

The SCC does not require symmetric transmit and receive clock signals—a feature allowing use of the wide variety of clock sources. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32, or 1/64 of the clock rate supplied to the receive and transmit clock inputs. In Asynchronous modes, the $\overline{\text{SYNC}}$ pin may be programmed as an input used for functions such as monitoring a ring indicator.

*Synchronous Modes.* The SCC supports both byte-oriented and bit-oriented synchronous communication. Synchronous byte-oriented protocols can be handled in several modes, allowing character synchronization with a 6-bit or 8-bit synchronous character (Monosync), any 12-bit synchronization pattern (Bisync), or with an external synchronous signal. Leading sync characters can be removed without interrupting the CPU.

Five- or 7-bit synchronous characters are detected with 8- or 16-bit patterns in the SCC by overlapping the larger pattern across multiple incoming synchronous characters as shown in Figure 4.

CRC checking for Synchronous byte-oriented modes is delayed by one character time so that the CPU may disable CRC checking on specific characters. This permits the implementation of protocols such as IBM Bisync.

Both CRC-16 ($X^{16} + X^{15} + X^2 + 1$) and CCITT ($X^{16} + X^{12} + X^5 + 1$) error checking polynomials are supported. Either polynomial may be selected in all Synchronous modes. Users may preset the CRC generator and checker to all 1s or all 0s. The SCC also provides a feature that automatically transmits CRC data when no other data is available for transmission. This allows for high speed transmissions under DMA control, with no need for CPU intervention at the end of a message. When there is no data or CRC to send in Synchronous modes, the transmitter inserts 6-, 8-, or 16-bit synchronous characters, regardless of the programmed character length.

The SCC supports Synchronous bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero insertion, and CRC generation. A special command can be used to abort a frame in transmission. At the end of a message, the SCC automatically transmits the CRC and trailing flag when the transmitter underruns. The transmitter may also be programmed to send an idle line consisting of continuous flag characters or a steady marking condition.

If a transmit underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change so that an abort may be issued. The SCC may also be programmed to send an abort itself in case of an underrun, relieving the CPU of this task. One to eight bits per character can be sent, allowing reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically acquires synchronization on the leading flag of a frame in SDLC or HDLC and provides a synchronization signal on the $\overline{\text{SYNC}}$ pin (an interrupt can also be programmed). The receiver can be programmed to search for frames addressed by a single byte (or four bits within a byte) of a user-selected address or to a global broadcast address. In this mode, frames not matching either the user-selected or broadcast address are ignored. The number of address bytes can be extended under software control. For receiving data, an interrupt on the first received character, or an interrupt on every character, or on special condition only (end-of-frame) can be selected. The receiver automatically deletes all 0s inserted by the transmitter during character assembly. CRC is also calculated and is automatically checked to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers. In SDLC mode, the SCC must be programmed to use the SDLC CRC polynomial, but the generator and checker may be preset to all 1s or all 0s.

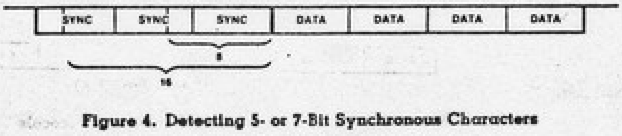


**Figure 4. Detecting 5- or 7-Bit Synchronous Characters**

## Functional Description (Continued)

The CRC is inverted before transmission and the receiver checks against the bit pattern 0001110100001111.

NRZ, NRZI or FM coding may be used in any 1x mode. The parity options available in Asynchronous modes are available in Synchronous modes.

The SCC can be conveniently used under DMA control to provide high speed reception or transmission. In reception, for example, the SCC can interrupt the CPU when the first character of a message is received. The CPU then enables the DMA to transfer the message to memory. The SCC then issues an end-of-frame interrupt and the CPU can check the status of the received message. Thus, the CPU is freed for other service while the message is being received. The CPU may also enable the DMA first and have the SCC interrupt only on end-of-frame. This procedure allows all data to be transferred via the DMA.

**SDLC Loop Mode.** The SCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that manages the message traffic flow on the loop and any number of secondary stations. In SDLC Loop mode, the SCC performs the functions of a secondary station while an SCC operating in regular SDLC mode can act as a controller (Figure 5).

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop, and in fact must pass these messages to the rest of the loop by retransmitting them with a one-bit-time delay. The secondary station can place its own message on the loop only at specific times. The controller signals that secondary stations may transmit messages by sending a special character, called an EOP (End Of Poll), around the loop. The EOP character is the bit pattern 11111110. Because of zero insertion during messages, this bit pattern is unique and easily recognized.

When a secondary station has a message to transmit and recognizes an EOP on the line, it changes the last binary 1 of the EOP to a 0 before transmission. This has the effect of turning the EOP into a flag sequence. The secondary station now places its message on the loop and terminates the message with an EOP. Any secondary stations further down the loop with messages to transmit can then append their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send merely echo the incoming messages and are prohibited from placing messages on the loop (except upon recognizing an EOP).

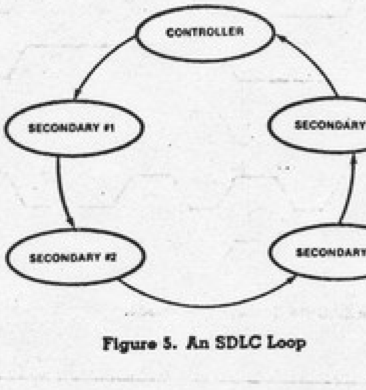


**Figure 5. An SDLC Loop**

SDLC Loop mode is a programmable option in the SCC. NRZ, NRZI, and FM coding may all be used in SDLC Loop mode.

**Baud Rate Generator.** Each channel in the SCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On startup, the flip-flop on the output is set in a High state, the value in the time constant register is loaded into the counter, and the counter starts counting down. The output of the baud rate generator toggles upon reaching 0, the value in the time constant register is loaded into the counter, and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both. It can also drive the Digital Phase-Locked Loop (see next section).

If the receive clock or transmit clock is not programmed to come from the $\overline{\text{TRxC}}$ pin, the output of the baud rate generator may be echoed out via the $\overline{\text{TRxC}}$ pin.

The following formula relates the time constant to the baud rate (the baud rate is in bits/second and the BR clock period is in seconds):

$$\text{baud rate} = \frac{1}{2\,(\text{time constant} + 2) \times (\text{BR clock period})}$$

**Digital Phase-Locked Loop.** The SCC contains a Digital Phase-Locked-Loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding. The DPLL is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock may then be used as the SCC receive clock, the transmit clock, or both.

For NRZI encoding, the DPLL counts the 32x clock to create nominal bit times. As the 32x clock is counted, the DPLL is searching the

**Functional Description** (Continued)

incoming data stream for edges (either 1 to 0 or 0 to 1). Whenever an edge is detected, the DPLL makes a count adjustment (during the next counting cycle), producing a terminal count closer to the center of the bit cell.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15 to 16 counting transition.

The 32x clock for the DPLL can be programmed to come from either the $\overline{\text{RTxC}}$ input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the SCC via the $\overline{\text{TRxC}}$ pin (if this pin is not being used as an input).

**Data Encoding.** The SCC may be programmed to encode and decode the serial data in four different ways (Figure 6). In NRZ encoding, a 1 is represented by a High level and a 0 is represented by a Low level. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level. In FM1 (more properly, bi-phase mark), a transition occurs at the beginning of every bit cell. A 1 is represented by an additional transition at the center of the bit cell and a 0 is represented by no additional transition at the center of the bit cell. In FM0 (bi-phase space), a transition occurs at the beginning of every bit cell. A 0 is represented by an additional transition at the center of the bit cell, and a 1 is represented by no additional transition at the center of the bit cell. In addition to these four methods, the SCC can be used to decode Manchester (bi-phase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0 to 1, the bit is a 0. If the transition is 1 to 0, the bit is a 1.

**Auto Echo and Local Loopback.** The SCC is capable of automatically echoing everything it receives. This feature is useful mainly in Asynchronous modes, but works in Synchronous and SDLC modes as well. In Auto Echo mode, TxD is RxD. Auto Echo mode can be used with NRZI or FM encoding with no additional delay, because the data stream is not decoded before retransmission. In Auto Echo mode, the $\overline{\text{CTS}}$ input is ignored as a transmitter enable (although transitions on this input can still cause interrupts if programmed to do so). In this mode, the transmitter is actually bypassed and the programmer is responsible for disabling transmitter interrupts and $\overline{\text{WAIT/REQUEST}}$ on transmit.

The SCC is also capable of local loopback. In this mode TxD is RxD, just as in Auto Echo mode. However, in Local Loopback mode, the internal transmit data is tied to the internal receive data and RxD is ignored (except to be echoed out via TxD). The $\overline{\text{CTS}}$ and $\overline{\text{DCD}}$ inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works in Asynchronous, Synchronous and SDLC modes with NRZ, NRZI or FM coding of the data stream.

**I/O Interface Capabilities.** The SCC offers the choice of Polling, Interrupt (vectored or nonvectored), and Block Transfer modes to transfer data, status, and control information to and from the CPU. The Block Transfer mode can be implemented under CPU or DMA control.

**Polling.** All interrupts are disabled. Three status registers in the SCC are automatically updated whenever any function is performed. For example, end-of-frame in SDLC mode sets a bit in one of these status registers. The idea behind polling is for the CPU to periodically read a status register until the register contents indicate the need for data to be transferred. Only one register needs to be

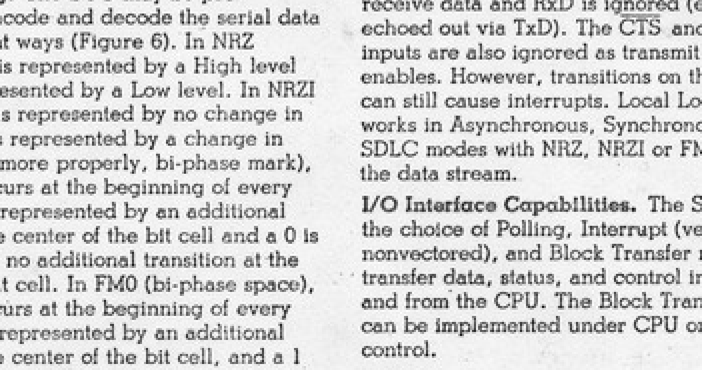


**Figure 6. Data Encoding Methods**

**Functional Description** (Continued)

read; depending on its contents, the CPU either writes data, reads data, or continues. Two bits in the register indicate the need for data transfer. An alternative is a poll of the Interrupt Pending register to determine the source of an interrupt. The status for both channels resides in one register.

**Interrupts.** When an SCC responds to an Interrupt Acknowledge signal ($\overline{\text{INTACK}}$) from the CPU, an interrupt vector may be placed on the data bus. This vector is written in WR2 and may be read in RR2A or RR2B (Figures 10 and 11).

To speed interrupt response time, the SCC can modify three bits in this vector to indicate status. If the vector is read in Channel A, status is never included; if it is read in Channel B, status is always included.

Each of the six sources of interrupts in the SCC (Transmit, Receive, and External/Status interrupts in both channels) has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). Operation of the IE bit is straightforward. If the IE bit is set for a given interrupt source, then that source can request interrupts. The exception is when the MIE (Master Interrupt Enable) bit in WR9 is reset and no interrupts may be requested. The IE bits are write only.

The other two bits are related to the interrupt priority chain (Figure 7). As a microprocessor peripheral, the SCC may request an interrupt only when no higher priority device is requesting one, e.g., when IEI is High. If the device in question requests an interrupt, it pulls down $\overline{\text{INT}}$. The CPU then responds with $\overline{\text{INTACK}}$, and the interrupting device places the vector on the data bus.

In the SCC, the IP bit signals a need for interrupt servicing. When an IP bit is 1 and the IEI input is High, the $\overline{\text{INT}}$ output is pulled Low, requesting an interrupt. In the SCC, if the IE bit is not set by enabling interrupts, then the IP for that source can never be set. The IP bits are readable in RR3A.

The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the SCC and external to the SCC are prevented from requesting interrupts. The internal interrupt sources are inhibited by the state of the internal daisy chain, while lower priority devices are inhibited by the IEO output of the SCC being pulled Low and propagated to subsequent peripherals. An IUS bit is set during an Interrupt Acknowledge cycle if there are no higher priority devices requesting interrupts.

There are three types of interrupts: Transmit, Receive, and External/Status. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receiver, Transmit, and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. (This implies that the transmitter must have had a data character written into it so that it can become empty.) When enabled, the receiver can interrupt the CPU in one of three ways:

- Interrupt on First Receive Character or Special Receive Condition.
- Interrupt on All Receive Characters or Special Receive Condition.
- Interrupt on Special Receive Condition Only.

Interrupt on First Character or Special Condition and Interrupt on Special Condition Only are typically used with the Block Transfer mode. A Special Receive Condition is one of the following: receiver overrun, framing error in Asynchronous mode, end-of-frame in SDLC mode and, optionally, a parity error. The Special Receive Condition interrupt is different from an ordinary receive character available interrupt only in the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt can occur from Special Receive Conditions any time after the first receive character interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the $\overline{\text{CTS}}$, $\overline{\text{DCD}}$, and $\overline{\text{SYNC}}$ pins; however, an

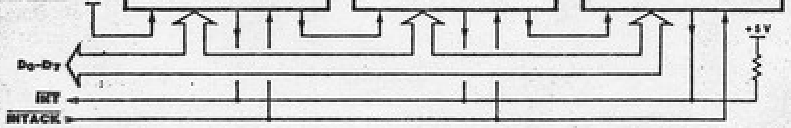


**Figure 7. Interrupt Schedule**

## Functional Description (Continued)

External/Status interrupt is also caused by a Transmit Underrun condition, or a zero count in the baud rate generator, or by the detection of a Break (Asynchronous mode), Abort (SDLC mode) or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the SCC to interrupt when the Abort or EOP sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Abort condition in external logic in SDLC mode. In SDLC Loop mode, this feature allows secondary stations to recognize the wishes of the primary station to regain control of the loop during a poll sequence.

**CPU/DMA Block Transfer.** The SCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode uses the $\overline{\text{WAIT}}$/$\overline{\text{REQUEST}}$ output in conjunction with the Wait/Request bits in WR1. The $\overline{\text{WAIT}}$/$\overline{\text{REQUEST}}$ output can be defined under software control as a $\overline{\text{WAIT}}$ line in the CPU Block Transfer mode or as a $\overline{\text{REQUEST}}$ line in the DMA Block Transfer mode.

To a DMA controller, the SCC $\overline{\text{REQUEST}}$ output indicates that the SCC is ready to transfer data to or from memory. To the CPU, the $\overline{\text{WAIT}}$ line indicates that the SCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The $\overline{\text{DTR}}$/$\overline{\text{REQUEST}}$ line allows full-duplex operation under DMA control.

## Architecture

The SCC internal structure includes two full-duplex channels, two baud rate generators, internal control and interrupt logic, and a bus interface to a nonmultiplexed bus. Associated with each channel are a number of read and write registers for mode control and status information, as well as logic necessary to interface to modems or other external devices (Figure 8).

The logic for both channels provides formats, synchronization, and validation for data transferred to and from the channel interface. The modem control inputs are monitored by the control logic under program control. All of the modem control signals are general-purpose in nature and can optionally be used for functions other than modem control.

The register set for each channel includes ten control (write) registers, two sync-character (write) registers, and four status (read) registers. In addition, each baud rate generator has two (read/write) registers for holding the time constant that determines the baud rate. Finally, associated with the interrupt logic is a write register for the interrupt vector accessible through either channel, a

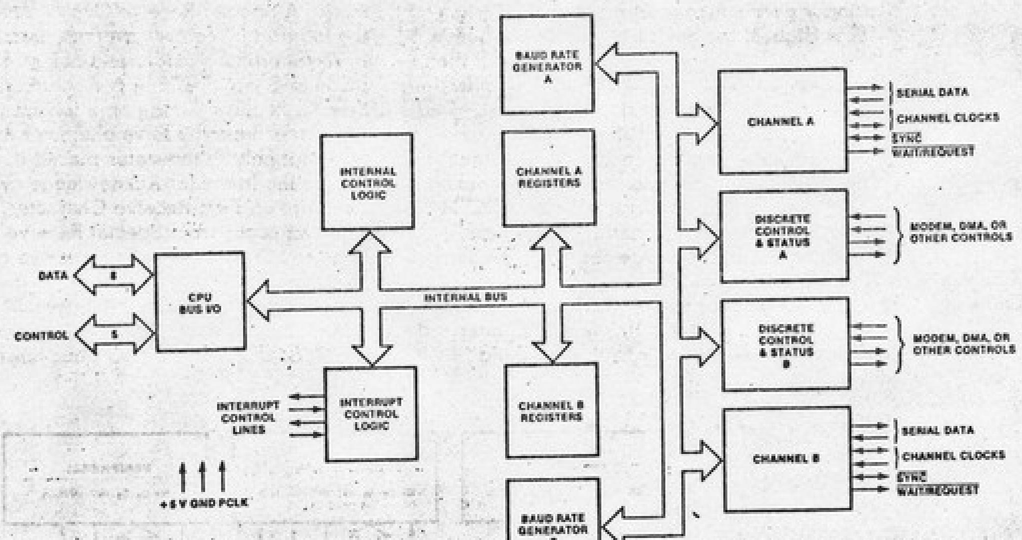


**Figure 8. Block Diagram of SCC Architecture**

## Architecture
(Continued)

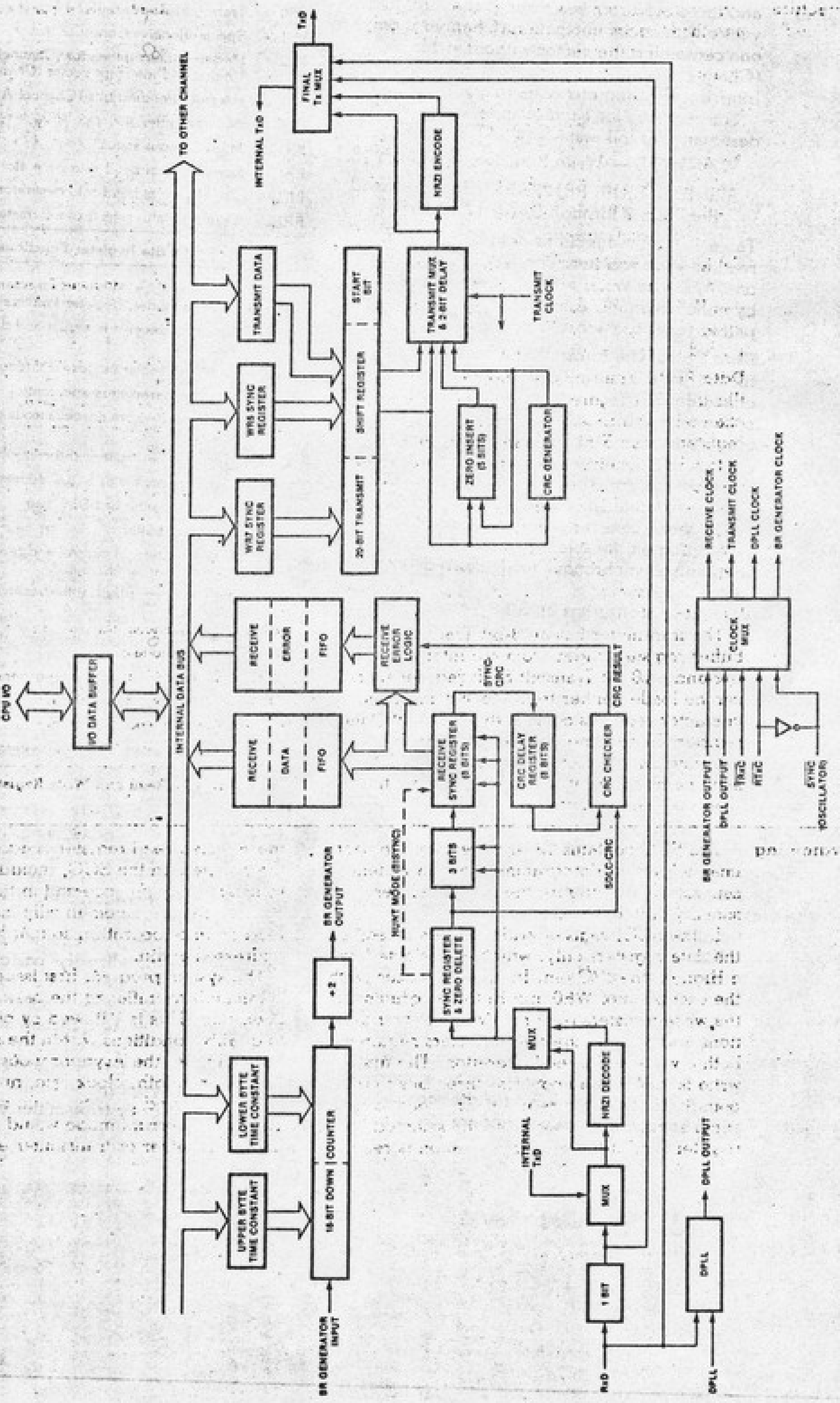


Figure 8. Data Path

## Architecture (Continued)

write only Master Interrupt Control register and three read registers: one containing the vector with status infomation (Channel B only), one containing the vector without status (Channel A only), and one containing the Interrupt Pending bits (Channel A only).

The registers for each channel are designated as follows:

WR0–WR15 — Write Registers 0 through 15.

RR0–RR3, RR10, RR12, RR13, RR15 — Read Registers 0 through 3, 10, 12, 13, 15.

Table 1 lists the functions assigned to each read or write register. The SCC contains only one WR2 and WR9, but they can be accessed by either channel. All other registers are paired (one for each channel).

**Data Path.** The transmit and receive data path illustrated in Figure 9 is identical for both channels. The receiver has three 8-bit buffer registers in an FIFO arrangement, in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service an interrupt at the beginning of a block of high speed data. Incoming data is routed through one of several paths (data or CRC) depending on the selected mode (the character length in Asynchronous modes also determines the data path).

The transmitter has an 8-bit Transmit Data buffer register loaded from the internal data bus and a 20-bit Transmit Shift register that can be loaded either from the synchronous character registers or from the Transmit Data register. Depending on the operational mode, outgoing data is routed through one of four main paths before it is transmitted from the Transmit Data output (TxD)

| **Read Register Functions** | |
|---|---|
| RR0 | Transmit/Receive buffer status and External status |
| RR1 | Special Receive Condition status |
| RR2 | Modified interrupt vector (Channel B only)<br>Unmodified interrupt vector (Channel A only) |
| RR3 | Interrupt Pending bits (Channel A only) |
| RR8 | Receive buffer |
| RR10 | Miscellaneous status |
| RR12 | Lower byte of baud rate generator time constant |
| RR13 | Upper byte of baud rate generator time constant |
| RR15 | External/Status interrupt information |
| **Write Register Functions** | |
| WR0 | CRC initialize, initialization commands for the various modes, Register Pointers |
| WR1 | Transmit/Receive interrupt and data transfer mode definition |
| WR2 | Interrupt vector (accessed through either channel) |
| WR3 | Receive parameters and control |
| WR4 | Transmit/Receive miscellaneous parameters and modes |
| WR5 | Transmit parameters and controls |
| WR6 | Sync characters or SDLC address field |
| WR7 | Sync character or SDLC flag |
| WR8 | Transmit buffer |
| WR9 | Master interrupt control and reset (accessed through either channel) |
| WR10 | Miscellaneous transmitter/receiver control bits |
| WR11 | Clock mode control |
| WR12 | Lower byte of baud rate generator time constant |
| WR13 | Upper byte of baud rate generator time constant |
| WR14 | Miscellaneous control bits |
| WR15 | External/Status interrupt control |

**Table 1. Read and Write Register Functions**

## Programming

The SCC contains 13 write registers in each channel that are programmed by the system separately to configure the functional personality of the channels.

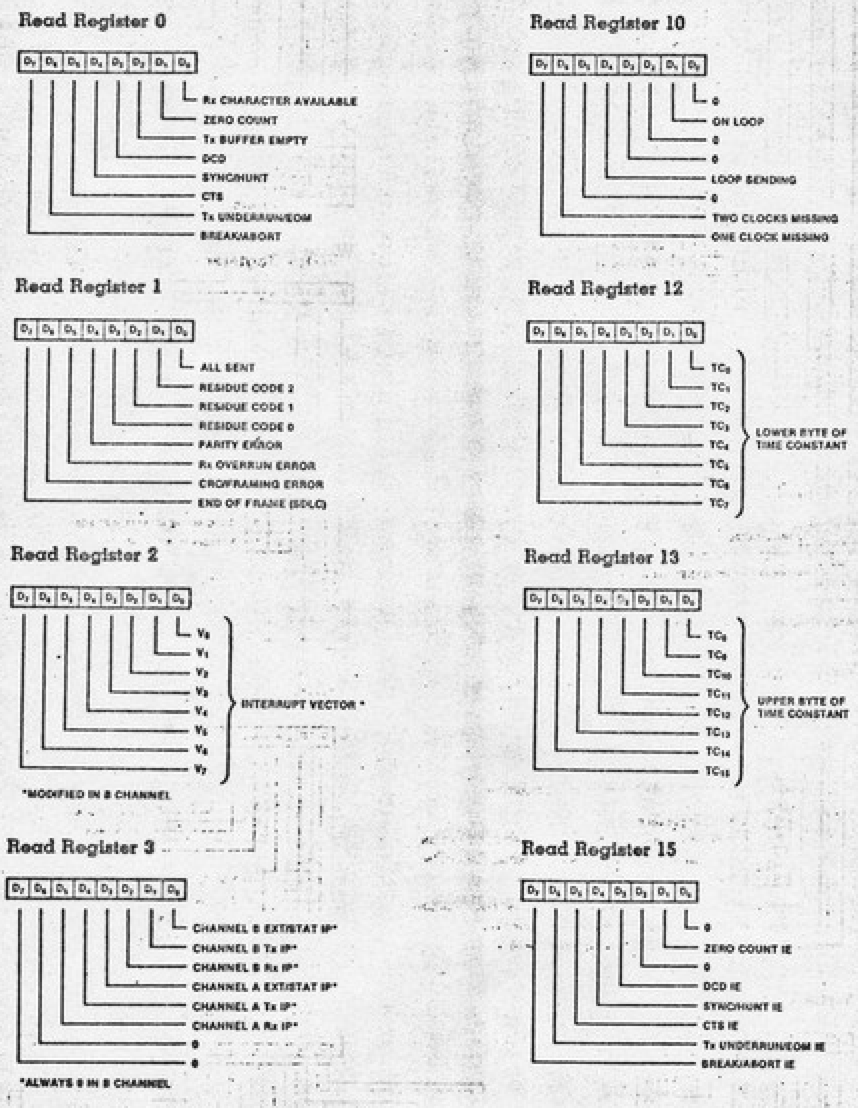
In the SCC, register addressing is direct for the data registers only, which are selected by a High on the D/$\overline{C}$ pin. In all other cases (with the exception of WR0 and RR0), programming the write registers requires two write operations and reading the read registers requires both a write and a read operation. The first write is to WR0 and contains three bits that point to the selected register. The second write is the actual control word for the selected register, and if the second operation is read, the selected read register is accessed. All of the registers in the SCC, including the data registers, may be accessed in this fashion. The pointer bits are automatically cleared after the read or write operation so that WR0 (or RR0) is addressed again.

The system program first issues a series of commands to initialize the basic mode of operation. This is followed by other commands to qualify conditions within the selected mode. For example, the Asynchronous mode, character length, clock rate, number of stop bits, even or odd parity might be set first. Then the interrupt mode would be set, and finally, receiver or transmitter enable.

**Programming (Continued)**

**Read Registers.** The SCC contains eight read registers (actually nine, counting the receive buffer (RR8) in each channel). Four of these may be read to obtain status information (RR0, RR1, RR10, and RR15). Two registers (RR12 and RR13) may be read to learn the baud rate generator time constant. RR2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information (Channel B). RR3 contains the Interrupt Pending (IP) bits (Channel A). Figure 10 shows the formats for each read register.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring; e.g., when the interrupt vector indicates a Special Receive Condition interrupt, all the appropriate error bits can be read from a single register (RR1).



Read Register 0

Read Register 10

Read Register 1

Read Register 12

Read Register 2

Read Register 13

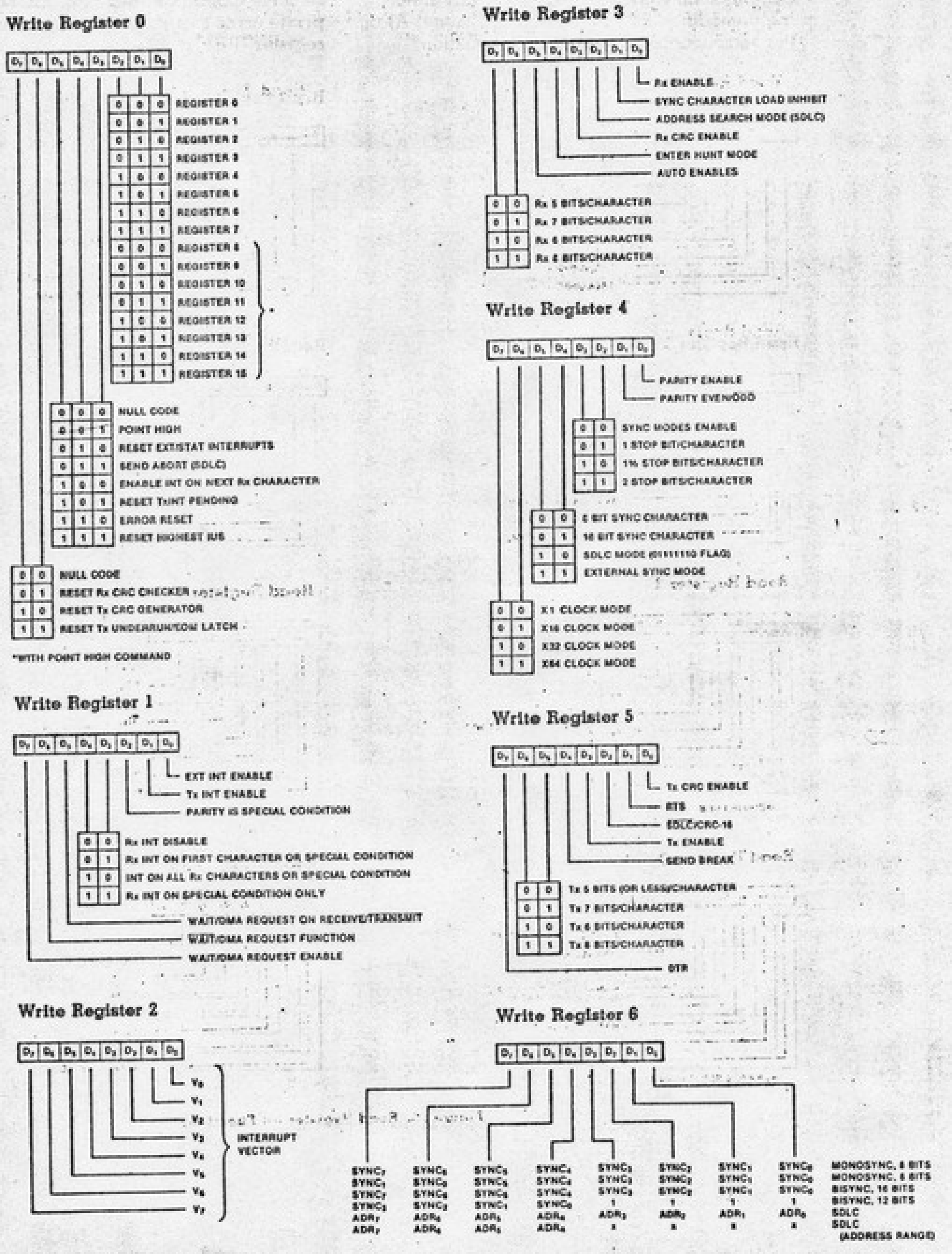
Read Register 3

Read Register 15

**Figure 10. Read Register Bit Functions**

**Programming (Continued)**

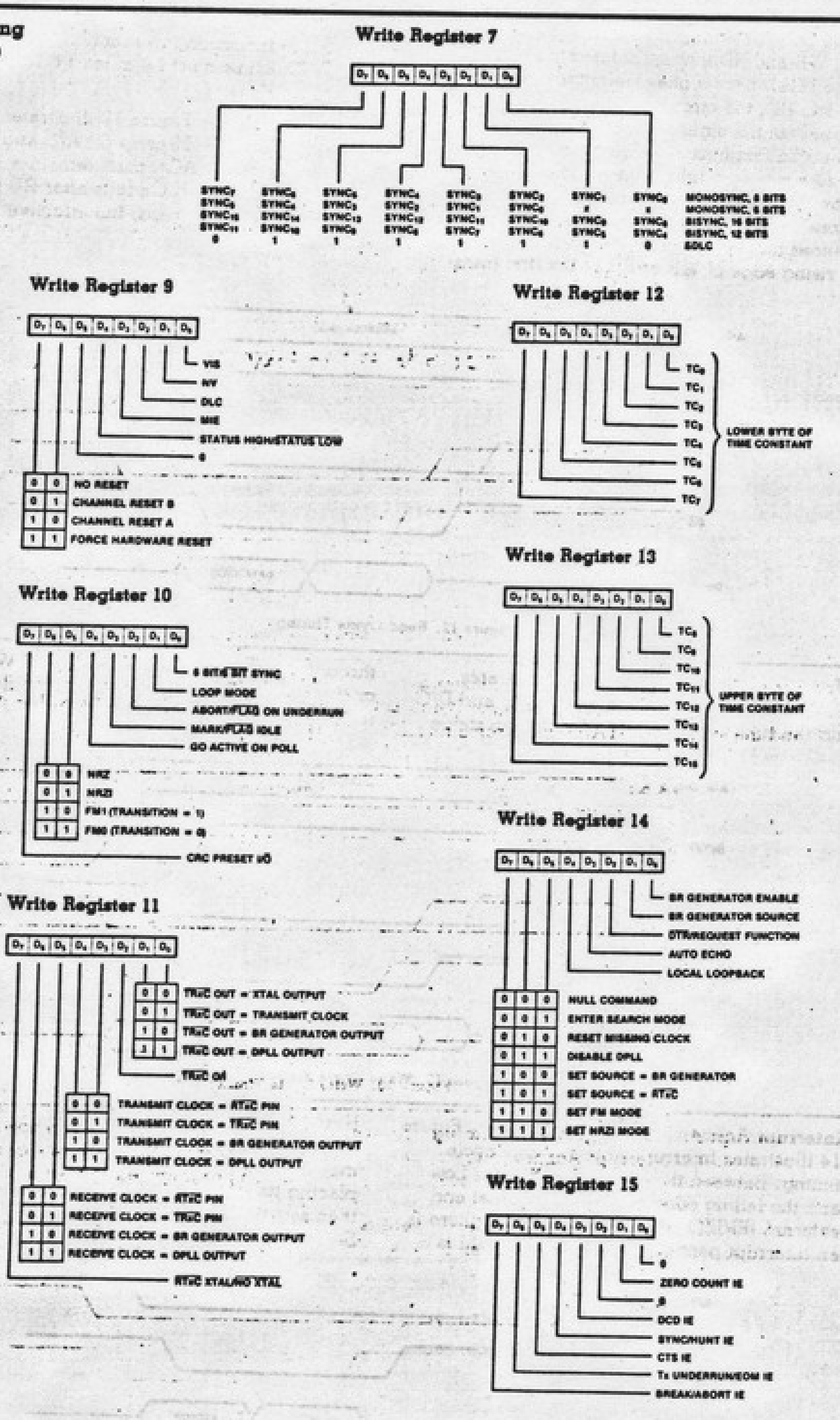
**Write Registers.** The SCC contains 13 write registers (14 counting WR8, the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. In addition, there are two registers (WR2 and WR9) shared by the two channels that may be accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. Figure 11 shows the format of each write register.



| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | |
|---|---|---|---|---|---|---|---|---|
| $SYNC_7$ | $SYNC_6$ | $SYNC_5$ | $SYNC_4$ | $SYNC_3$ | $SYNC_2$ | $SYNC_1$ | $SYNC_0$ | MONOSYNC, 8 BITS |
| $SYNC_1$ | $SYNC_0$ | $SYNC_5$ | $SYNC_4$ | $SYNC_3$ | $SYNC_2$ | $SYNC_1$ | $SYNC_0$ | MONOSYNC, 6 BITS |
| $SYNC_7$ | $SYNC_6$ | $SYNC_5$ | $SYNC_4$ | $SYNC_3$ | $SYNC_2$ | $SYNC_1$ | $SYNC_0$ | BISYNC, 16 BITS |
| $SYNC_3$ | $SYNC_2$ | $SYNC_1$ | $SYNC_0$ | 1 | 1 | 1 | 1 | BISYNC, 12 BITS |
| $ADR_7$ | $ADR_6$ | $ADR_5$ | $ADR_4$ | $ADR_3$ | $ADR_2$ | $ADR_1$ | $ADR_0$ | SDLC |
| $ADR_7$ | $ADR_6$ | $ADR_5$ | $ADR_4$ | x | x | x | x | SDLC (ADDRESS RANGE) |

**Figure 11. Write Register Bit Functions**

2016-006 2023-010



**Write Register 7**

D7 D6 D5 D4 D3 D2 D1 D0
SYNC7 SYNC5 SYNC15 SYNC11 0
SYNC6 SYNC4 SYNC14 SYNC10 1
SYNC5 SYNC3 SYNC13 SYNC9 1
SYNC4 SYNC2 SYNC12 SYNC8 1
SYNC3 SYNC1 SYNC11 SYNC7 1
SYNC2 SYNC0 SYNC10 SYNC6 1
SYNC1 x SYNC9 SYNC5 1
SYNC0 x SYNC8 SYNC4 0
MONOSYNC, 8 BITS
MONOSYNC, 6 BITS
BISYNC, 16 BITS
BISYNC, 12 BITS
SDLC

**Write Register 9**

**Write Register 10**

**Write Register 11**

D7 D6 D5 D4 D3 D2 D1 D0
0 0 TRxC OUT = XTAL OUTPUT
0 1 TRxC OUT = TRANSMIT CLOCK
1 0 TRxC OUT = BR GENERATOR OUTPUT
1 1 TRxC OUT = DPLL OUTPUT
TRxC O/I
0 0 TRANSMIT CLOCK = RTxC PIN
0 1 TRANSMIT CLOCK = TRxC PIN
1 0 TRANSMIT CLOCK = BR GENERATOR OUTPUT
1 1 TRANSMIT CLOCK = DPLL OUTPUT
0 0 RECEIVE CLOCK = RTxC PIN
0 1 RECEIVE CLOCK = TRxC PIN
1 0 RECEIVE CLOCK = BR GENERATOR OUTPUT
1 1 RECEIVE CLOCK = DPLL OUTPUT
RTxC XTAL/NO XTAL

**Write Register 12**

**Write Register 13**

**Write Register 14**

**Write Register 15**

**Figure 11. Write Register Bit Functions** (Continued)

## Timing

The SCC generates internal control signals from $\overline{WR}$ and $\overline{RD}$ that are related to PCLK. Since PCLK has no phase relationship with $\overline{WR}$ and $\overline{RD}$, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the SCC. The recovery time required for proper operation is specified from the rising edge of $\overline{WR}$ or $\overline{RD}$ in the first transaction involving the SCC to the falling edge of $\overline{WR}$ or $\overline{RD}$ in the second transaction involving the SCC. This time must be at least 6 PCLK cycles plus 200 ns.

**Read Cycle Timing.** Figure 12 illustrates Read cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{RD}$ falls or if it rises before $\overline{RD}$ rises, the effective $\overline{RD}$ is shortened.

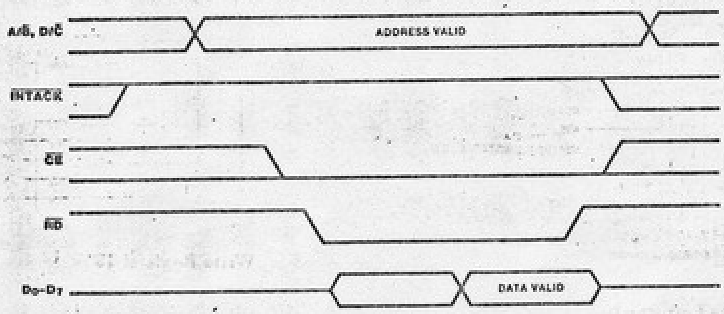


**Figure 12. Read Cycle Timing**

**Write Cycle Timing.** Figure 13 illustrates Write cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{WR}$ falls or if it rises before $\overline{WR}$ rises, the effective $\overline{WR}$ is shortened.

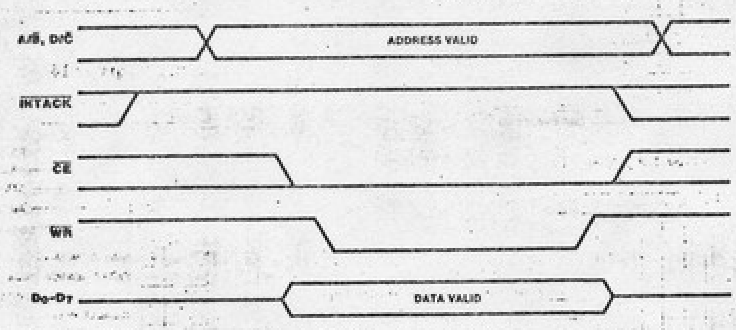


**Figure 13. Write Cycle Timing**

**Interrupt Acknowledge Cycle Timing.** Figure 14 illustrates Interrupt Acknowledge cycle timing. Between the time $\overline{INTACK}$ goes Low and the falling edge of $\overline{RD}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the SCC and IEI is High when $\overline{RD}$ falls, the Acknowledge cycle is intended for the SCC. In this case, the SCC may be programmed to respond to $\overline{RD}$ Low by placing its interrupt vector on $D_0$–$D_7$ and it then sets the appropriate Interrupt-Under-Service latch internally.

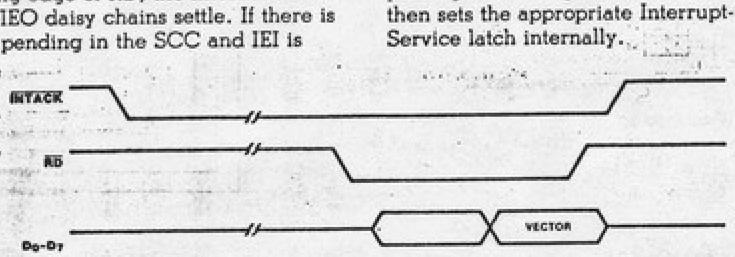


**Figure 14. Interrupt Acknowledge Cycle Timing**

**Absolute Maximum Ratings**

Voltages on all inputs and outputs
with respect to GND..........-0.3 V to +7.0 V
Operating Ambient
Temperature..................As Specified in
Ordering Information
Storage Temperature........-65°C to +150°C

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Standard Test Conditions**

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

- $+4.75\ V \le V_{CC} \le +5.25\ V$
- GND = 0 V
- $T_A$ as specified in Ordering Information

All ac parameters assume a load capacitance of 50 pF max.

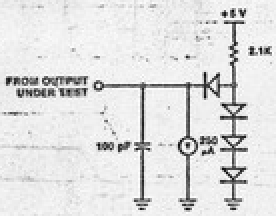


**Figure 15. Standard Test Load**

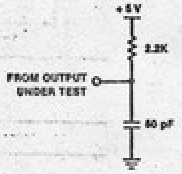


**Figure 16. Open-Drain Test Load**

Z8530 SCC

**DC Characteristics**

| Symbol | Parameter | Min | Max | Unit | Condition |
|---|---|---|---|---|---|
| $V_{IH}$ | Input High Voltage | 2.0 | $V_{CC}+0.3$ | V | |
| $V_{IL}$ | Input Low Voltage | -0.3 | 0.8 | V | |
| $V_{OH}$ | Output High Voltage | 2.4 | | V | $I_{OH} = -250\ \mu A$ |
| $V_{OL}$ | Output Low Voltage | | 0.4 | V | $I_{OL} = +2.0\ mA$ |
| $I_{IL}$ | Input Leakage | | ±10.0 | µA | $0.4 \le V_{IN} \le +2.4V$ |
| $I_{OL}$ | Output Leakage | | ±10.0 | µA | $0.4 \le V_{OUT} \le +2.4V$ |
| $I_{CC}$ | $V_{CC}$ Supply Current | | 250 | mA | |

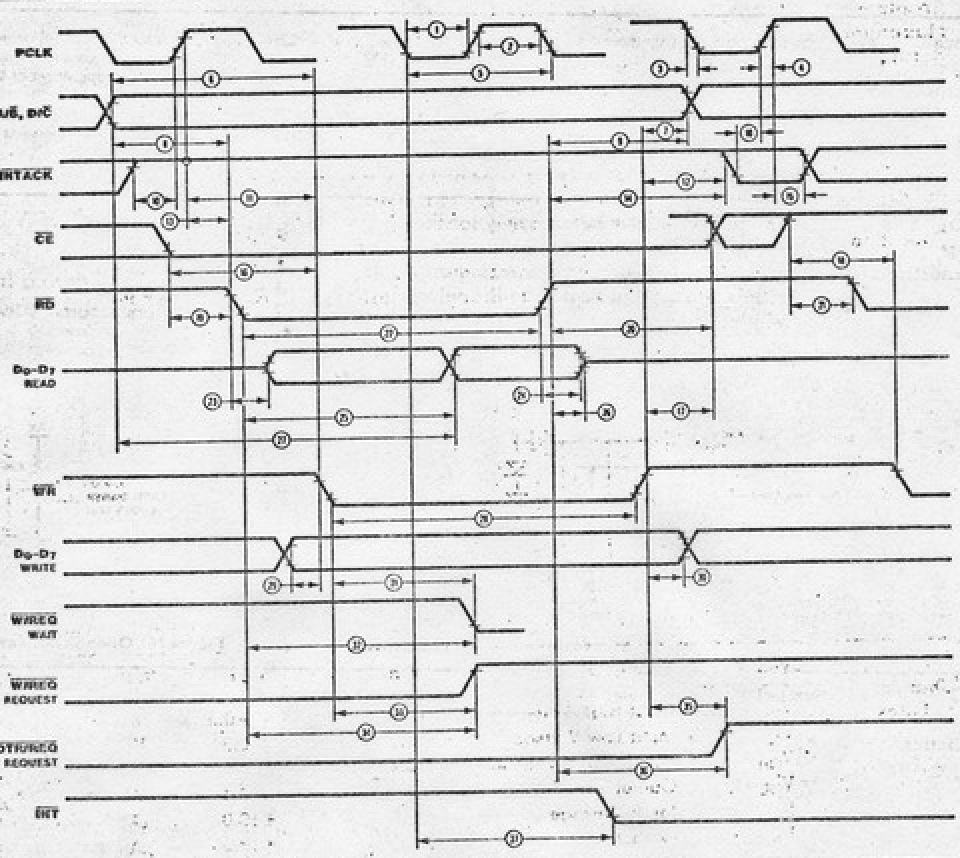
$V_{CC}$ = 5 V ± 5% unless otherwise specified, over specified temperature range.

**Capacitance**

| Symbol | Parameter | Min | Max | Unit | Test Condition |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance | | 10 | pF | Unmeasured Pins Returned to Ground |
| $C_{OUT}$ | Output Capacitance | | 15 | pF | |
| $C_{I/O}$ | Bidirectional Capacitance | | 20 | pF | |

f = 1 MHz, over specified temperature range.

## Read and Write Timing



| No. | Symbol | Parameter | 4 MHz Min | 4 MHz Max | 6 MHz Min | 6 MHz Max | Notes*† |
|---|---|---|---|---|---|---|---|
| 1 | TwPCl | PCLK Low Width | 105 | 2000 | 70 | 1000 | |
| 2 | TwPCh | PCLK High Width | 105 | 2000 | 70 | 1000 | |
| 3 | TfPC | PCLK Fall Time | | 20 | | 10 | |
| 4 | TrPC | PCLK Rise Time | | 20 | | 15 | |
| 5 | TcPC | PCLK Cycle Time | 250 | 4000 | 165 | 2000 | |
| 6 | TsA(WR) | Address to $\overline{WR}$ ↓ Setup Time | 80 | | 80 | | |
| 7 | ThA(WR) | Address to $\overline{WR}$ ↑ Hold Time | 0 | | 0 | | |
| 8 | TsA(RD) | Address to $\overline{RD}$ ↓ Setup Time | 80 | | 80 | | |
| 9 | ThA(RD) | Address to $\overline{RD}$ ↑ Hold Time | 0 | | 0 | | |
| 10 | TsIA(PC) | $\overline{INTACK}$ to PCLK ↑ Setup Time | 0 | | 0 | | |
| 11 | TsIAi(WR) | $\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time | 200 | | 160 | | 1 |
| 12 | ThIA(WR) | $\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time | 0 | | 0 | | |
| 13 | TsIAi(RD) | $\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time | 200 | | 160 | | 1 |
| 14 | ThIA(RD) | $\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time | 0 | | 0 | | |
| 15 | ThIA(PC) | $\overline{INTACK}$ to PCLK ↑ Hold Time | 100 | | 100 | | |
| 16 | TsCEl(WR) | $\overline{CE}$ Low to $\overline{WR}$ ↓ Setup Time | 0 | | 0 | | |
| 17 | ThCE(WR) | $\overline{CE}$ to $\overline{WR}$ ↑ Hold Time | 0 | | 0 | | |
| 18 | TsCEh(WR) | $\overline{CE}$ High to $\overline{WR}$ ↓ Setup Time | 100 | | 70 | | |
| 19 | TsCEl(RD) | $\overline{CE}$ Low to $\overline{RD}$ ↓ Setup Time | 0 | | 0 | | 1 |
| 20 | ThCE(RD) | $\overline{CE}$ to $\overline{RD}$ ↑ Hold Time | 0 | | 0 | | 1 |
| 21 | TsCEh(RD) | $\overline{CE}$ High to $\overline{RD}$ ↓ Setup Time | 100 | | 70 | | 1 |
| 22 | TwRDl | $\overline{RD}$ Low Width | 390 | | 250 | | 1 |
| 23 | TdRD(DRA) | $\overline{RD}$ ↓ to Read Data Active Delay | 0 | | 0 | | |
| 24 | TdRDr(DR) | $\overline{RD}$ ↑ to Read Data Not Valid Delay | 0 | | 0 | | |
| 25 | TdRDf(DR) | $\overline{RD}$ ↓ to Read Data Valid Delay | | 250 | | 180 | |
| 26 | TdRD(DRz) | $\overline{RD}$ ↑ to Read Data Float Delay | | 70 | | 45 | 2 |

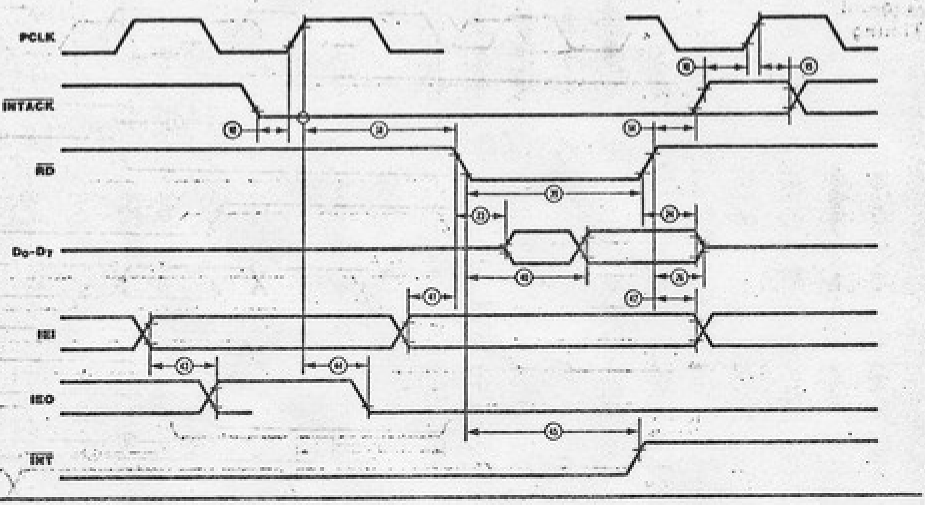
NOTES:

1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time required for a ±0.5 V change in the output with a maximum dc load and minimum ac load.

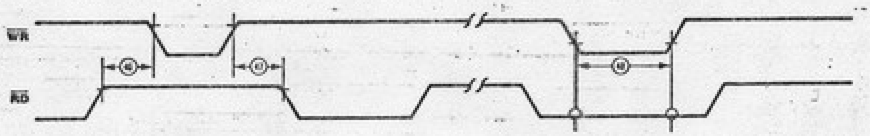
* Timings are preliminary and subject to change.

† Units in nanoseconds (ns).

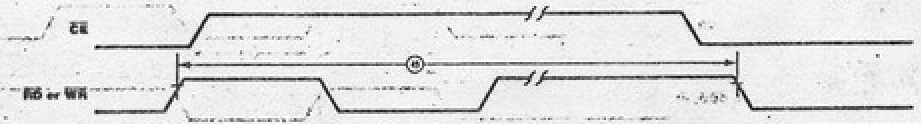
**Interrupt Acknowledge Timing**



**Reset Timing**



**Cycle Timing**

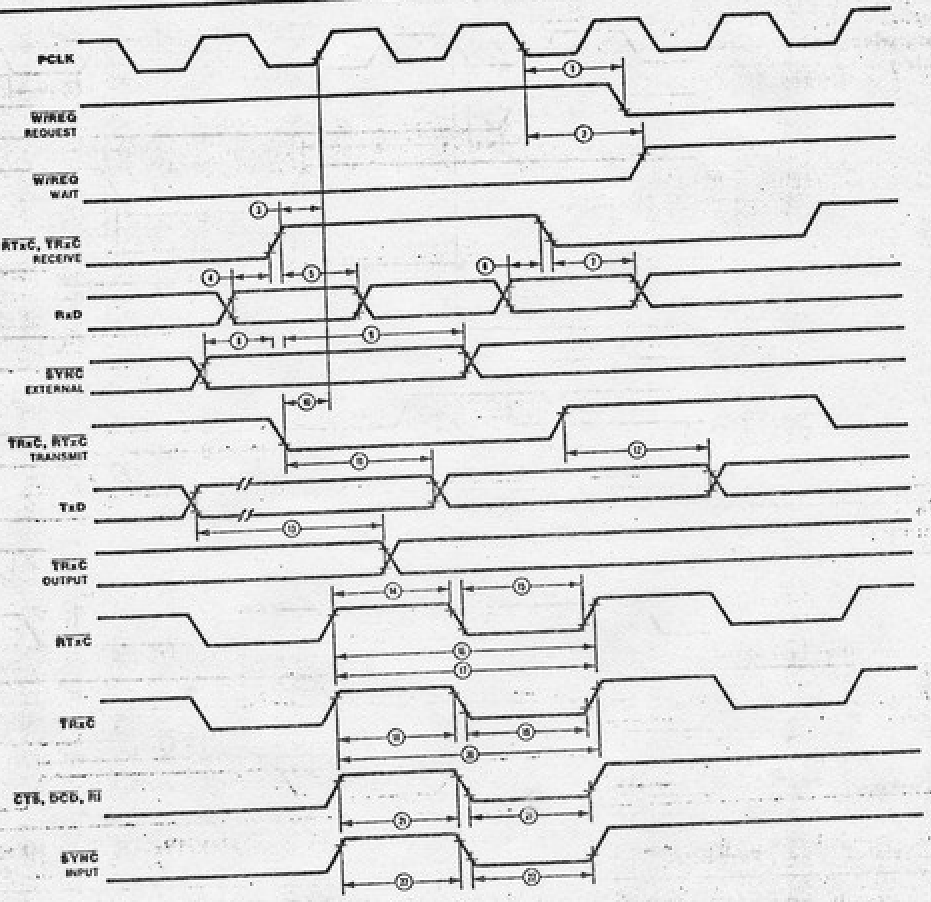


| No. | Symbol | Parameter | 4 MHz Min | 4 MHz Max | 6 MHz Min | 6 MHz Max | Notes*† |
|---|---|---|---|---|---|---|---|
| 27 | TdA(DR) | Address Required Valid to Read Data Valid Delay | | 590 | | 420 | |
| 28 | TwWRl | $\overline{\text{WR}}$ Low Width | 390 | | 250 | | |
| 29 | TsDW(WR) | Write Data to $\overline{\text{WR}}$ ↓ Setup Time | 0 | | 0 | | |
| 30 | ThDW(WR) | Write Data to $\overline{\text{WR}}$ ↑ Hold Time | 0 | | 0 | | |
| 31 | TdWR(W) | $\overline{\text{WR}}$ ↓ to Wait Valid Delay | | 240 | | 200 | 4 |
| 32 | TdRD(W) | $\overline{\text{RD}}$ ↓ to Wait Valid Delay | | 240 | | 200 | 4 |
| 33 | TdWRf(REQ) | $\overline{\text{WR}}$ ↓ to $\overline{\text{W/REQ}}$ Not Valid Delay | | 240 | | 200 | |
| 34 | TdRDf(REQ) | $\overline{\text{RD}}$ ↓ to $\overline{\text{W/REQ}}$ Not Valid Delay | | 240 | | 200 | |
| 35 | TdWRr(REQ) | $\overline{\text{WR}}$ ↑ to $\overline{\text{DTR/REQ}}$ Not Valid Delay | | 5TcPC<br>+300 | | 5TcPC<br>+250 | |
| 36 | TdRDr(REQ) | $\overline{\text{RD}}$ ↑ to $\overline{\text{DTR/REQ}}$ Not Valid Delay | | 5TcPC<br>+300 | | 5TcPC<br>+250 | |
| 37 | TdPC(INT) | PCLK ↓ to $\overline{\text{INT}}$ Valid Delay | | 500 | | 500 | 4 |
| 38 | TdIAi(RD) | $\overline{\text{INTACK}}$ to $\overline{\text{RD}}$ ↓ (Acknowledge) Delay | 250 | | 250 | | 5 |
| 39 | TwRDA | $\overline{\text{RD}}$ (Acknowledge) Width | 285 | | 250 | | |
| 40 | TdRDA(DR) | $\overline{\text{RD}}$ ↓ (Acknowledge) to Read Data Valid Delay | | 190 | | 180 | |
| 41 | TsIEI(RDA) | IEI to $\overline{\text{RD}}$ ↓ (Acknowledge) Setup Time | 120 | | 100 | | |
| 42 | ThIEI(RDA) | IEI to $\overline{\text{RD}}$ ↑ (Acknowledge) Hold Time | 0 | | 0 | | |
| 43 | TdIEI(IEO) | IEI to IEO Delay Time | | 120 | | 100 | |
| 44 | TdPC(IEO) | PCLK ↑ to IEO Delay | | 250 | | 250 | |
| 45 | TdRDA(INT) | $\overline{\text{RD}}$ ↓ to $\overline{\text{INT}}$ Inactive Delay | | 500 | | 500 | 4 |
| 46 | TdRD(WRQ) | $\overline{\text{RD}}$ ↑ to $\overline{\text{WR}}$ ↓ Delay for No Reset | 30 | | 15 | | |
| 47 | TdWRQ(RD) | $\overline{\text{WR}}$ ↑ to $\overline{\text{RD}}$ ↓ Delay for No Reset | 30 | | 30 | | |
| 48 | TwRES | $\overline{\text{WR}}$ and $\overline{\text{RD}}$ Coincident Low for Reset | 250 | | 250 | | |
| 49 | Trc | Valid Access Recovery Time | 6TcPC<br>+200 | | 6TcPC<br>+130 | | 3 |

NOTES:
3. Parameter applies only between transactions involving the SCC.
4. Open-drain output, measured with open-drain test load.
5. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) ≤ 250 + for the highest priority device in the daisy chain, TsIEI(RDA) ≤ 110 = 360 for the SCC, and TdIEI(IEO) for each device separating them in the daisy chain.

* Timings are preliminary and subject to change.
† Units in nanoseconds (ns).

## General Timing



2016-014

| No. | Symbol | Parameter | 4 MHz |  | 6 MHz |  | Notes*† |
|---|---|---|---|---|---|---|---|
|  |  |  | Min | Max | Min | Max |  |
| 1 | TdPC(REQ) | PCLK ↓ to $\overline{\text{W/REQ}}$ Valid Delay |  | 250 |  | 250 |  |
| 2 | TdPC(W) | PCLK ↓ to Wait Inactive Delay |  | 350 |  | 350 |  |
| 3 | TsRXC(PC) | $\overline{\text{RxC}}$ ↑ to PCLK ↑ Setup Time (PCLK ÷ 4 case only) | 80 | TwPCl | 70 | TwPCl | 1,4 |
| 4 | TsRXD(RXCr) | RxD to $\overline{\text{RxC}}$ ↑ Setup Time (X1 Mode) | 0 |  | 0 |  | 1 |
| 5 | ThRXD(RXCr) | RxD to $\overline{\text{RxC}}$ ↑ Hold Time (X1 Mode) | 150 |  | 150 |  | 1 |
| 6 | TsRXD(RXCf) | RxD to $\overline{\text{RxC}}$ ↓ Setup Time (X1 Mode) | 0 |  | 0 |  | 1,5 |
| 7 | ThRXD(RXCf) | RxD to $\overline{\text{RxC}}$ ↓ Hold Time (X1 Mode) | 150 |  | 150 |  | 1,5 |
| 8 | TsSY(RXC) | $\overline{\text{SYNC}}$ to $\overline{\text{RxC}}$ ↑ Setup Time | -200 |  | -200 |  | 1 |
| 9 | ThSY(RXC) | $\overline{\text{SYNC}}$ to $\overline{\text{RxC}}$ ↑ Hold Time | 3TcPC<br>+200 |  | 3TcPC<br>+200 |  | 1 |
| 10 | TsTXC(PC) | $\overline{\text{TxC}}$ ↓ to PCLK ↑ Setup Time | 0 |  | 0 |  | 2,4 |
| 11 | TdTXCf(TXD) | $\overline{\text{TxC}}$ ↓ to TxD Delay (X1 Mode) |  | 300 |  | 230 | 2 |
| 12 | TdTXCr(TXD) | $\overline{\text{TxC}}$ ↑ to TxD Delay (X1 Mode) |  | 300 |  | 230 | 2,5 |
| 13 | TdTXD(TRX) | TxD to $\overline{\text{TRxC}}$ Delay (Send Clock Echo) |  | 200 |  | 200 |  |
| 14 | TwRTXh | $\overline{\text{RTxC}}$ High Width | 180 |  | 180 |  | 6 |
| 15 | TwRTXl | $\overline{\text{RTxC}}$ Low Width | 180 |  | 180 |  | 6 |
| 16 | TcRTX | $\overline{\text{RTxC}}$ Cycle Time | 400 |  | 400 |  | 6 |
| 17 | TcRTXX | Crystal Oscillator Period | 250 | 1000 | 250 | 1000 | 3 |
| 18 | TwTRXh | $\overline{\text{TRxC}}$ High Width | 180 |  | 180 |  | 6 |
| 19 | TwTRXl | $\overline{\text{TRxC}}$ Low Width | 180 |  | 180 |  | 6 |
| 20 | TcTRX | $\overline{\text{TRxC}}$ Cycle Time | 400 |  | 400 |  | 6 |
| 21 | TwEXT | $\overline{\text{DCD}}$ or $\overline{\text{CTS}}$ Pulse Width | 200 |  | 200 |  |  |
| 22 | TwSY | $\overline{\text{SYNC}}$ Pulse Width | 200 |  | 200 |  |  |

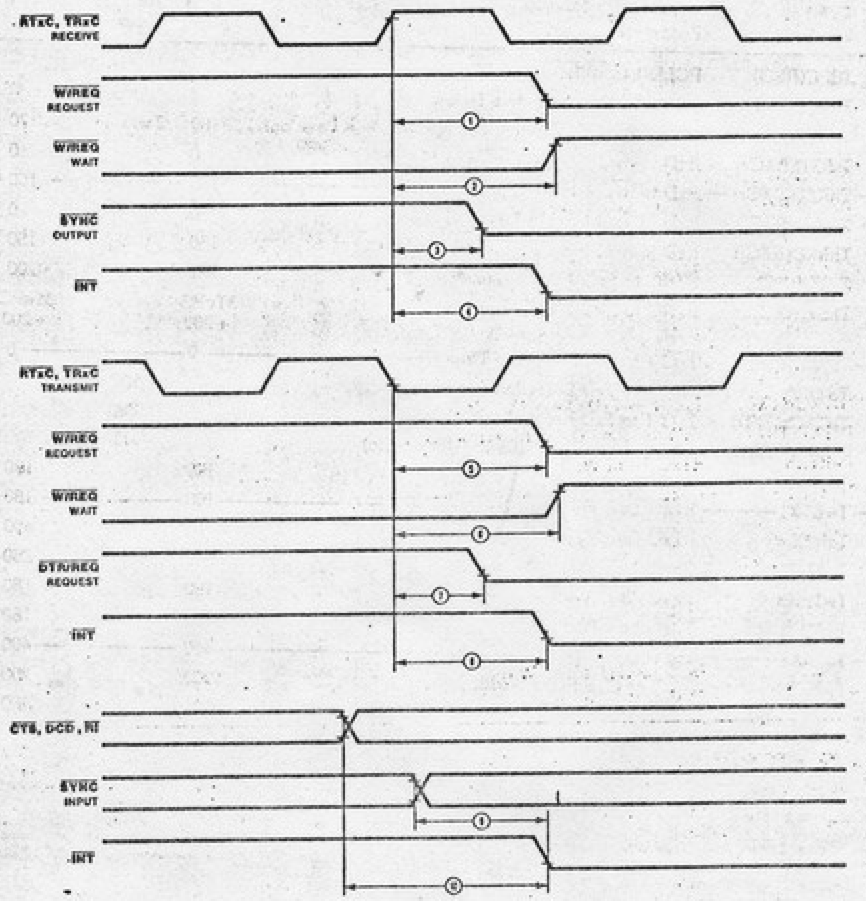
NOTES:

1. $\overline{\text{RxC}}$ is $\overline{\text{RTxC}}$ or $\overline{\text{TRxC}}$, whichever is supplying the receive clock.
2. $\overline{\text{TxC}}$ is $\overline{\text{TRxC}}$ or $\overline{\text{RTxC}}$, whichever is supplying the transmit clock.
3. Both $\overline{\text{RTxC}}$ and $\overline{\text{SYNC}}$ have 30 pF capacitors to ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between $\overline{\text{RxC}}$ and PCLK or $\overline{\text{TxC}}$ and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver; DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.

* Timings are preliminary and subject to change.

† Units in nanoseconds (ns).

## System Timing



| No. | Symbol | Parameter | 4 MHz Min | 4 MHz Max | 6 MHz Min | 6 MHz Max | Notes*† |
|---|---|---|---|---|---|---|---|
| 1 | TdRXC(REQ) | $\overline{RxC}$ ↑ to $\overline{W/REQ}$ Valid Delay | 8 | 12 | 8 | 12 | 2 |
| 2 | TdRXC(W) | $\overline{RxC}$ ↑ to Wait Inactive Delay | 8 | 12 | 8 | 12 | 1,2 |
| 3 | TdRXC(SY) | $\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay | 4 | 7 | 4 | 7 | 2 |
| 4 | TdRXC(INT) | $\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay | 10 | 16 | 10 | 16 | 1,2 |
| ~~5~~ | ~~TdTXC(REQ)~~ | ~~$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay~~ | ~~5~~ | ~~8~~ | ~~5~~ | ~~8~~ | ~~3~~ |
| 6 | TdTXC(W) | $\overline{TxC}$ ↓ to Wait Inactive Delay | 5 | 8 | 5 | 8 | 1,3 |
| 7 | TdTXC(DRQ) | $\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay | 4 | 7 | 4 | 7 | 3 |
| 8 | TdTXC(INT) | $\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay | 6 | 10 | 6 | 10 | 1,3 |
| 9 | TdSY(INT) | $\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay | 2 | 6 | 2 | 6 | 1 |
| 10 | TdEXT(INT) | $\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay | 2 | 6 | 2 | 6 | 1 |

NOTES:
1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.

* Timings are preliminary and subject to change.
† Units equal to TcPC.

## Ordering Information

| Product Number | Package/ Temp | Speed | Description | Product Number | Package/ Temp | Speed | Description |
|---|---|---|---|---|---|---|---|
| Z8530 | CE | 4.0 MHz | SCC (40-pin) | Z8530A | CE | 6.0 MHz | SCC (40-pin) |
| Z8530 | CM | 4.0 MHz | Same as above | Z8530A | CM | 6.0 MHz | Same as above |
| Z8530 | CMB | 4.0 MHz | Same as above | Z8530A | CMB | 6.0 MHz | Same as above |
| Z8530 | CS | 4.0 MHz | Same as above | Z8530A | CS | 6.0 MHz | Same as above |
| Z8530 | DE | 4.0 MHz | Same as above | Z8530A | DE | 6.0 MHz | Same as above |
| Z8530 | DS | 4.0 MHz | Same as above | Z8530A | DS | 6.0 MHz | Same as above |
| Z8530 | PE | 4.0 MHz | Same as above | Z8530A | PE | 6.0 MHz | Same as above |
| Z8530 | PS | 4.0 MHz | Same as above | Z8530A | PS | 6.0 MHz | Same as above |

NOTES: C = Ceramic, D = Cerdip, P = Plastic; E = -40°C to +85°C, M = -55°C to 125°C, MB = -55°C to 125°C with MIL-STD-883 with Class B processing, S = 0°C to +70°C.

Part No.: 32 958 044
Related doc: 3

# Z8536 CIO Counter/Timer and Parallel I/O Unit

Zilog

## Product Specification

September 1983

### Features

- Two independent 8-bit, double-buffered, bidirectional I/O ports plus a 4-bit special-purpose I/O port. I/O ports feature programmable polarity, programmable direction (Bit mode), "pulse catchers," and programmable open-drain outputs.
- Four handshake modes, including 3-Wire (like the IEEE-488).
- REQUEST/$\overline{\text{WAIT}}$ signal for high-speed data transfer.
- Flexible pattern-recognition logic, programmable as a 16-vector interrupt controller.
- Three independent 16-bit counter/timers with up to four external access lines per counter/timer (count input, output, gate, and trigger), and three output duty cycles (pulsed, one-shot, and square-wave), programmable as retriggerable or nonretriggerable.
- Easy to use since all registers are read/write.

### General Description

The Z8536 CIO Counter/Timer and Parallel I/O element is a general-purpose peripheral circuit, satisfying most counter/timer and parallel I/O needs encountered in system designs. This versatile device contains three I/O ports and three counter/timers. Many programmable options tailor its configuration to specific applications. The use of the device is simplified by making all internal registers (command, status, and data) readable and (except for status bits) writable. In addition, each register is given its own unique internal address, so that any register can be accessed in two operations. All data registers can be directly accessed in a single operation. The CIO is easily interfaced to all popular microprocessors.

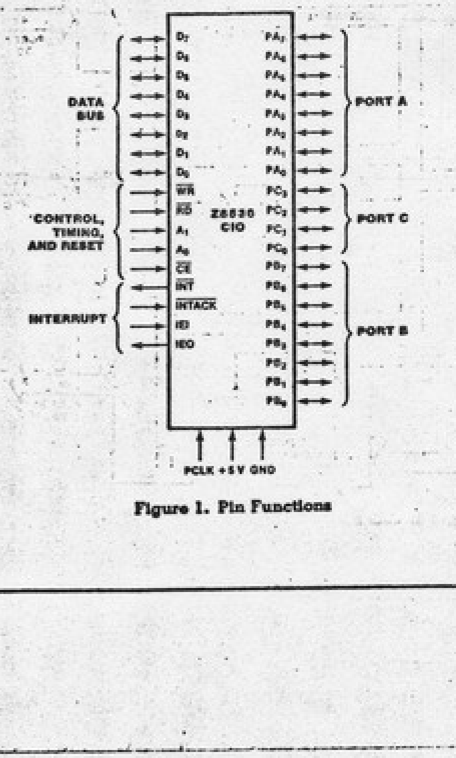


**Figure 1. Pin Functions**

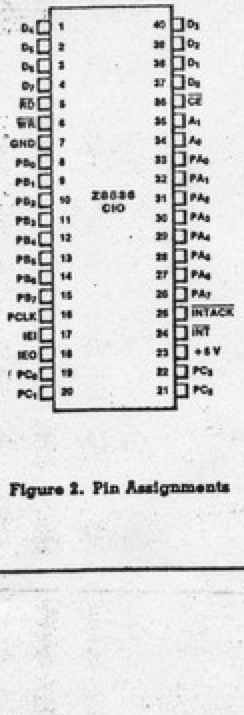


**Figure 2. Pin Assignments**

## Pin Description

**$A_0$–$A_1$.** *Address Lines* (input). These two lines are used to select the register involved in the CPU transaction: Port A's Data register, Port B's Data register, Port C's Data register, or a control register.

**CE.** *Chip Enable* (input, active Low). A Low level on this input enables the CIO to be read from or written to.

**$D_0$–$D_7$.** *Data Bus* (bidirectional 3-state). These eight data lines are used for transfers between the CPU and the CIO.

**IEI.** *Interrupt Enable In* (input, active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A High IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.

**IEO.** *Interrupt Enable Out* (output, active High). IEO is High only if IEI is High and the CPU is not servicing an interrupt from the requesting CIO or is not requesting an interrupt (Interrupt Acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.

**INT.** *Interrupt Request* (output, open-drain, active Low). This signal is pulled Low when the CIO requests an interrupt.

**INTACK.** *Interrupt Acknowledge* (input, active Low). This input indicates to the CIO that an Interrupt Acknowledge cycle is in progress. INTACK must be synchronized to PCLK, and it must be stable throughout the Interrupt Acknowledge cycle.

**$PA_0$–$PA_7$.** *Port A I/O lines* (bidirectional, 3-state, or open-drain). These eight I/O lines transfer information between the CIO's Port A and external devices.

**$PB_0$–$PB_7$.** *Port B I/O lines* (bidirectional, 3-state, or open-drain). These eight I/O lines transfer information between the CIO's Port B and external devices. May also be used to provide external access to Counter/Timers 1 and 2.

**$PC_0$–$PC_3$.** *Port C I/O lines* (bidirectional, 3-state, or open-drain). These four I/O lines are used to provide handshake, WAIT, and REQUEST lines for Ports A and B or to provide external access to Counter/Timer 3 or access to the CIO's Port C.

**PCLK.** *Peripheral Clock* (input, TTL-compatible). This is the clock used by the internal control logic and the counter/timers in timer mode. It does not have to be the CPU clock.

**RD*.** *Read* (input, active Low). This signal indicates that a CPU is reading from the CIO. During an Interrupt Acknowledge cycle, this signal gates the interrupt vector onto the data bus if the CIO is the highest priority device requesting an interrupt.

**WR*.** *Write* (input, active Low). This signal indicates a CPU write to the CIO.

*When RD and WR are detected Low at the same time (normally an illegal condition), the CIO is reset.

## Architecture

The CIO Counter/Timer and Parallel I/O element (Figure 3) consists of a CPU interface, three I/O ports (two general-purpose 8-bit ports and one special-purpose 4-bit port),

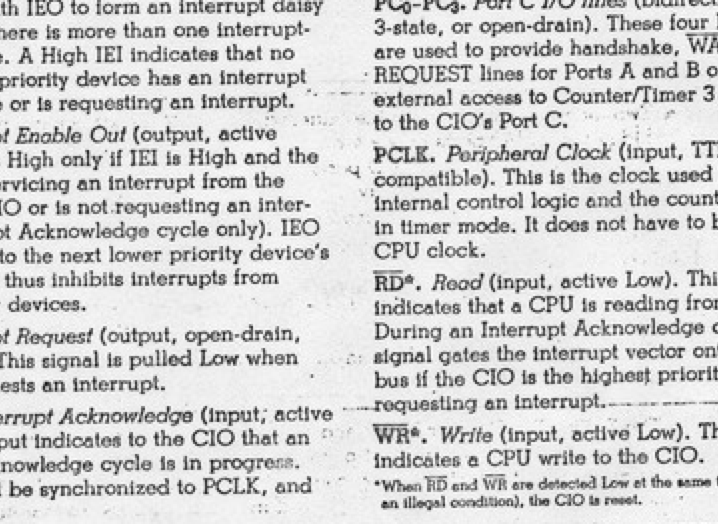


**Figure 3. CIO Block Diagram**

**Architecture**
(Continued)

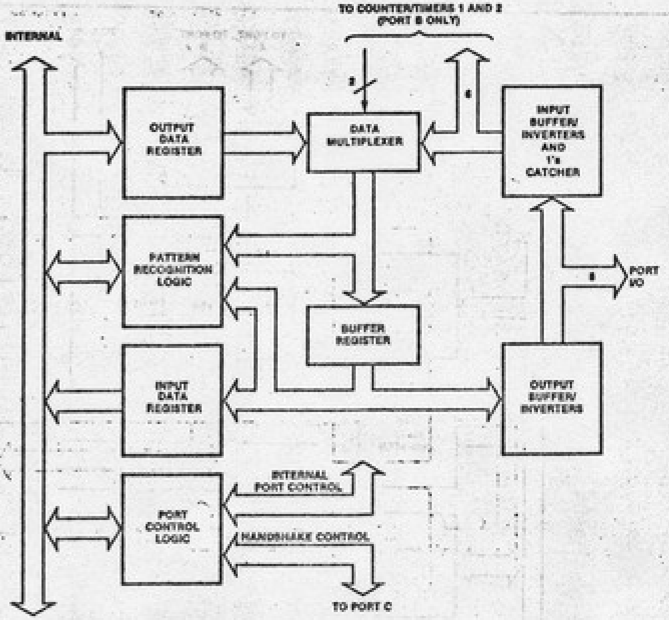


**Figure 4. Ports A and B Block Diagram**

three 16-bit counter/timers, an interrupt-control logic block, and the internal-control logic block. An extensive number of programmable options allow the user to tailor the configuration to best suit the specific application.

The two general-purpose 8-bit I/O ports (Figure 4) are identical, except that Port B can be specified to provide external access to Counter/Timers 1 and 2. Either port can be programmed to be a handshake-driven, double-buffered port (input, output, or bidirectional) or a control-type port with the direction of each bit individually programmable. Each port includes pattern-recognition logic, allowing interrupt generation when a specific pattern is detected. The pattern-recognition logic can be programmed so the port functions like a priority-interrupt controller. Ports A and B can also be linked to form a 16-bit I/O port.

To control these capabilities, both ports contain 12 registers. Three of these registers, the Input, Output, and Buffer registers, comprise the data path registers. Two registers, the Mode Specification and Handshake Specification registers, are used to define the mode of the port and to specify which handshake, if any, is to be used. The reference pattern for the pattern-recognition logic is defined via three registers: the Pattern Polarity, Pattern Transition, and Pattern Mask registers. The detailed characteristics of each bit path (for example, the direction of data flow or whether a path is inverting or noninverting) are programmed using the Data Path Polarity, Data Direction, and Special I/O Control registers.

The primary control and status bits are grouped in a single register, the Command and Status register, so that after the port is initially configured, only this register must be accessed frequently. To facilitate initialization, the port logic is designed so that registers associated with an unrequired capability are ignored and do not have to be programmed.

The function of the special-purpose 4-bit port, Port C (Figure 5), depends upon the roles of Ports A and B. Port C provides the required handshake lines. Any bits of Port C not used as handshake lines can be used as I/O lines or to provide external access for the third counter/timer.

Since Port C's function is defined primarily by Ports A and B, only three registers (besides the Data Input and Output registers) are needed. These registers specify the details of each bit path: the Data Path Polarity, Data Direction, and Special I/O Control registers.

**Architecture**
**(Continued)**

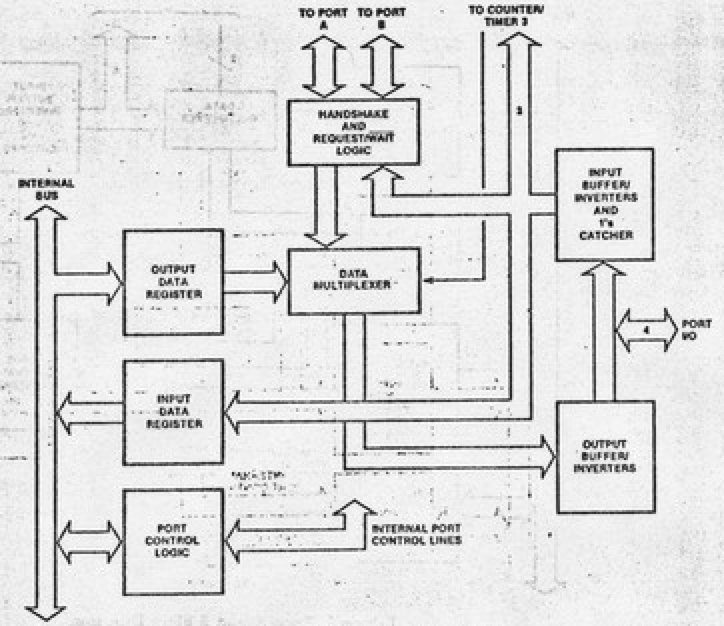


Figure 5. Port C Block Diagram

The three counter/timers (Figure 6) are all identical. Each is comprised of a 16-bit down-counter, a 16-bit Time Constant register (which holds the value loaded into the down-counter), a 16-bit Current Count register (used to read the contents of the down-counter), and two 8-bit registers for control and status (the Mode Specification and the Command and Status registers).

The capabilities of the counter/timer are numerous. Up to four port I/O lines can be dedicated as external access lines for each counter/timer: counter input, gate input, trigger input, and counter/timer output. Three different counter/timer output duty cycles are available: pulse, one-shot, or square-wave. The operation of the counter/timer can be programmed as either retriggerable or nonretriggerable. With these and other options, most counter/timer applications are covered.

There are five registers (Master Interrupt Control register, three Interrupt Vector registers, and the Current Vector register) associated with the interrupt logic. In addition, the ports' Command and Status registers and the counter/timers' Command and Status registers include bits associated with the interrupt logic. Each of these registers contains three bits for interrupt control and status: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE).

## Architecture
(Continued)

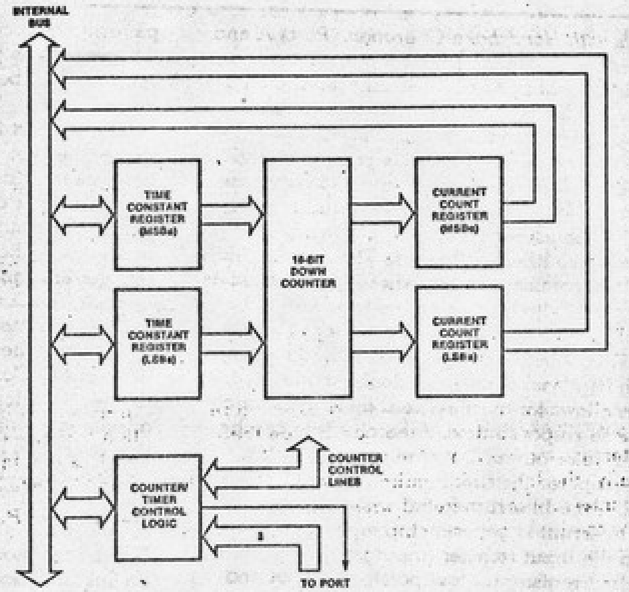

INTERNAL BUS
TIME CONSTANT REGISTER (MSBs)
TIME CONSTANT REGISTER (LSBs)
16-BIT DOWN COUNTER
CURRENT COUNT REGISTER (MSBs)
CURRENT COUNT REGISTER (LSBs)
COUNTER/ TIMER CONTROL LOGIC
COUNTER CONTROL LINES
3
TO PORT

**Figure 6. Counter/Timer Block Diagram**

## Functional Description

The following describes the functions of the ports, pattern-recognition logic, counter/timers, and interrupt logic.

**I/O Port Operations.** Of the CIO's three I/O ports, two (Ports A and B) are general-purpose, and the third (Port C) is a special-purpose 4-bit port. Ports A and B can be configured as input, output, or bidirectional ports with handshake. (Four different handshakes are available.) They can also be linked to form a single 16-bit port. If they are not used as ports with handshake, they provide 16 input or output bits with the data direction programmable on a bit-by-bit basis. Port B also provides access for Counter/Timers 1 and 2. In all configurations, Ports A and B can be programmed to recognize specific data patterns and to generate interrupts when the pattern is encountered.

The four bits of Port C provide the handshake lines for Ports A and B when required. A REQUEST/WAIT line can also be provided so that CIO transfers can be synchronized with DMAs or CPUs. Any Port C bits not used for handshake or REQUEST/WAIT can be used as input or output bits (individually data-direction programmable) or external access lines for Counter/Timer 3. Port C does not contain any pattern-recognition logic. It is, however, capable of bit-addressable writes. With this feature, any combination of bits can be set and/or cleared while the other bits remain undisturbed without first reading the register.

*Bit Port Operations.* In bit port operations, the port's Data Direction register specifies the direction of data flow for each bit. A 1 specifies an input bit, and a 0 specifies an output bit. If bits are used as I/O bits for a counter/timer, they should be set as input or output, as required.

The Data Path Polarity register provides the capability of inverting the data path. A 1 specifies inverting, and a 0 specifies noninverting. All discussions of the port operations assume that the path is noninverting.

The value returned when reading an input bit reflects the state of the input just prior to the read. A 1's catcher can be inserted into the input data path by programming a 1 to the corresponding bit position of the port's Special I/O Control register. When a 1 is detected at the 1's catcher input, its output is set to 1 until it is cleared. The 1's catcher is cleared by writing a 0 to the bit. In all other cases, attempted writes to input bits are ignored.

When Ports A and B include output bits, reading the Data register returns the value being output. Reads of Port C return the state of the pin. Outputs can be specified as open-drain by writing a 1 to the corresponding bit of the port's Special I/O Control register. Port C has the additional feature of bit-addressable writes. When writing to Port C, the four most significant bits are used as a write protect mask for the least significant bits (0–4, 1–5, 2–6, and 3–7). If the write protect bit is written with a 1, the state of the corresponding output bit is not changed.

## Functional Description (Continued)

*Ports with Handshake Operation.* Ports A and B can be specified as 8-bit input, output, or bidirectional ports with handshake. The CIO provides four different handshakes for its ports: Interlocked, Strobed, Pulsed, and 3-Wire. When specified as a port with handshake, the transfer of data into and out of the port and interrupt generation is under control of the handshake logic. Port C provides the handshake lines as shown in Table 1. Any Port C lines not used for handshake can be used as simple I/O lines or as access lines for Counter/Timer 3.

When Ports A and B are configured as ports with handshake, they are double-buffered. This allows for more relaxed interrupt service routine response time. A second byte can be input to or output from the port before the interrupt for the first byte is serviced. Normally, the Interrupt Pending (IP) bit is set and an interrupt is generated when data is shifted into the Input register (input port) of out of the Output register (output port). For input and output ports, the IP is automatically cleared when the data is read or written. In bidirectional ports, IP is cleared only by command. When the Interrupt on Two Bytes (ITB) control bit is set to 1, interrupts are generated only when two bytes of data are available to be read or written. This allows a minimum of 16 bits of information to be transferred on each interrupt. With ITB set, the IP is not automatically cleared until the second byte of data is read or written.

When the Single Buffer (SB) bit is set to 1, the port acts as if it is only single-buffered. This is useful if the handshake line must be stopped on a byte-by-byte basis.

Ports A and B can be linked to form a 16-bit port by programming a 1 in the Port Link Control (PLC) bit. In this mode, only Port A's Handshake Specification and Command and Status registers are used. Port B must be specified as a bit port. When linked, only Port A has pattern-match capability. Port B's pattern-match capability must be disabled. Also, when the ports are linked, Port B's Data register must be read or written before Port A's.

When a port is specified as a port with handshake, the type of port it is (input, output, or bidirectional) determines the direction of data flow. The data direction for the bidirectional port is determined by a bit in Port C (Table 1). In all cases, the contents of the Data Direction register are ignored. The contents of the Special I/O Control register apply only to output bits (3-state or open-drain). Inputs may not have 1's catchers; therefore, those bits in the Special I/O Control register are ignored. Port C lines used for handshake should be programmed as inputs. The handshake specification overrides Port C's Data Direction register for bits that must be outputs. The contents of Port C's Data Path Polarity register still apply.

**Interlocked Handshake.** In the Interlocked Handshake mode, the action of the CIO must be acknowledged by the external device before the next action can take place. Figure 7 shows timing for Interlocked Handshake. An output port does not indicate that new data is available until the external device indicates it is ready for the data. Similarly, an input port does not indicate that it is ready for new data until the data source indicates that the previous byte of the data is no longer available, thereby acknowledging the input port's acceptance of the last byte. This allows the CIO to interface directly to the port of a Z8 microcomputer, a UPC, an FIO, an FIFO, or to another CIO port with no external logic.

A 4-bit deskew timer can be inserted in the Data Available ($\overline{DAV}$) output for output ports. As data is transferred to the Buffer register, the deskew timer is triggered. After the number of PCLK cycles specified by the deskew timer time constant plus one, $\overline{DAV}$ is allowed to go Low. The deskew timer therefore guarantees that the output data is valid for a specified minimum amount of time before $\overline{DAV}$

| Port A/B Configuration | PC$_3$ | PC$_2$ | PC$_1$ | PC$_0$ |
|---|---|---|---|---|
| Ports A and B: Bit Ports | Bit I/O | Bit I/O | Bit I/O | Bit I/O |
| Port A: Input or Output Port (Interlocked, Strobed, or Pulsed Handshake)* | RFD or $\overline{DAV}$ | $\overline{ACKIN}$ | REQUEST/$\overline{WAIT}$ or Bit I/O | Bit I/O |
| Port B: Input or Output Port (Interlocked, Strobed, or Pulsed Handshake)* | REQUEST/$\overline{WAIT}$ or Bit I/O | Bit I/O | RFD or $\overline{DAV}$ | $\overline{ACKIN}$ |
| Port A or B: Input Port (3-Wire Handshake) | RFD (Output) | $\overline{DAV}$ (Input) | REQUEST/$\overline{WAIT}$ or Bit I/O | DAC (Output) |
| Port A or B: Output Port (3-Wire Handshake) | $\overline{DAV}$ (Output) | DAC (Input) | REQUEST/$\overline{WAIT}$ or Bit I/O | RFD (Input) |
| Port A or B: Bidirectional Port (Interlocked or Strobed Handshake) | RFD or $\overline{DAV}$ | $\overline{ACKIN}$ | REQUEST/$\overline{WAIT}$ or Bit I/O | IN/$\overline{OUT}$ |

*Both Ports A and B can be specified input or output with Interlocked, Strobed, or Pulsed Handshake at the same time if neither uses REQUEST/$\overline{WAIT}$.

**Table 1. Port C Bit Utilization**

**Functional Description** (Continued)

goes Low. Deskew timers are available for output ports independent of the type of handshake employed.

**Strobed Handshake.** In the Strobed Handshake mode, data is "strobed" into or out of the port by the external logic. The falling edge of the Acknowledge Input ($\overline{\text{ACKIN}}$) strobes data into or out of the port. Figure 7 shows timing for the Strobed Handshake. In contrast to the Interlocked handshake, the signal indicating the port is ready for another data transfer operates independently of the $\overline{\text{ACKIN}}$ input. It is up to the external logic to ensure that data overflows or underflows do not occur.

**3-Wire Handshake.** The 3-Wire Handshake is designed for the situation in which one output port is communicating with many input ports simultaneously. It is essentially the same as the Interlocked Handshake, except that two signals are used to indicate if an input port is ready for new data or if it has accepted the present data. In the 3-Wire Handshake (Figure 8), the rising edge of one status line indicates that the port is ready for data, and the rising edge of another status line indicates that the data has been accepted. With the 3-Wire Handshake, the output lines of many input ports can be bussed together with open-drain drivers; the output port knows when all the ports have accepted the data and are ready. This is the same handshake as is used on the IEEE-488 bus. Because this handshake requires three lines, only one port (either A or B) can be a 3-Wire Handshake port at a time. The 3-Wire Handshake is not available in the bidirectional mode. Because the port's direction can be changed under software control, however, bidirectional IEEE-488-type transfers can be performed.

**Pulsed Handshake.** The Pulsed Handshake (Figure 9) is designed to interface to mechanical-type devices that require data to be held for long periods of time and need relatively wide pulses to gate the data into or out of the device. The logic is the same as the Interlocked Handshake mode, except that an internal counter/timer is linked to the handshake logic. If the port is specified in the input mode, the timer is inserted in the $\overline{\text{ACKIN}}$ path. The external $\overline{\text{ACKIN}}$ input triggers the timer and its output is used as the Interlocked Handshake's normal acknowledge input. If the port is an output port, the timer is placed in the Data Available ($\overline{\text{DAV}}$) output path. The timer is triggered when the normal Interlocked Handshake $\overline{\text{DAV}}$ output goes Low and the timer output is used as the actual $\overline{\text{DAV}}$ output. The counter/timer maintains all of its normal capabilities. This handshake is not available to bidirectional ports.

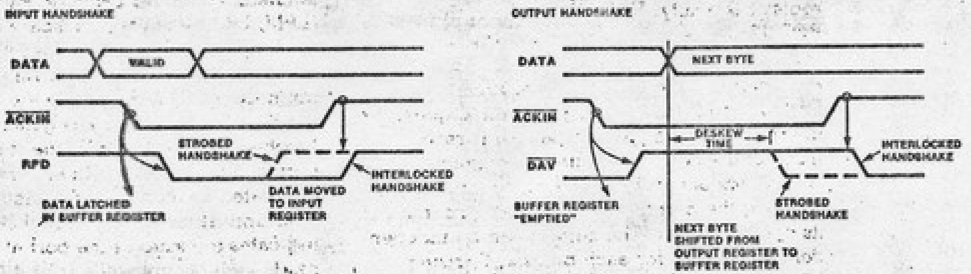


**Figure 7. Interlocked and Strobed Handshakes**

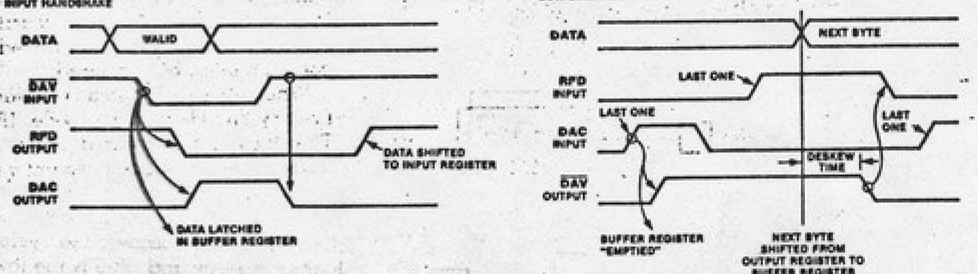


**Figure 8. 3-Wire Handshake**

**Functional Description (Continued)**

**REQUEST/$\overline{\text{WAIT}}$ Line Operation.** Port C can be programmed to provide a status signal output in addition to the normal handshake lines for either Port A or B when used as a port with handshake. The additional signal is either a REQUEST or $\overline{\text{WAIT}}$ signal. The REQUEST signal indicates when a port is ready to perform a data transfer via the CPU interface. It is intended for use with a DMA-type device. The $\overline{\text{WAIT}}$ signal provides synchronization for transfers with a CPU. Three bits in the Port Handshake Specification register provide controls for the REQUEST/$\overline{\text{WAIT}}$ logic. Because the extra Port C line is used, only one port can be specified as a port with a handshake and a REQUEST/$\overline{\text{WAIT}}$ line. The other port must be a bit port.

Operation of the REQUEST line is modified by the state of the port's Interrupt on Two Bytes (ITB) control bit. When ITB is 0, the REQUEST line goes active as soon as the CIO is ready for a data transfer. If ITB is 1, REQUEST does not go active until two bytes can be transferred. REQUEST stays active as long as a byte is available to be read or written.

The SPECIAL REQUEST function is reserved for use with bidirectional ports only. In this case, the REQUEST line indicates the status of the register not being used in the data path at that time. If the IN/$\overline{\text{OUT}}$ line is High, the REQUEST line is High when the Output register is empty. If IN/$\overline{\text{OUT}}$ is Low, the REQUEST line is High when the Input register is full.

*Pattern-Recognition Logic Operation.* Both Ports A and B can be programmed to generate interrupts when a specific pattern is recognized at the port. The pattern-recognition logic is independent of the port application, thereby allowing the port to recognize patterns in all of its configurations. The pattern can be independently specified for each bit as 1, 0, rising edge, falling edge, or any transition. Individual bits may be masked off. A pattern-match is defined as the simultaneous satisfaction of all nonmasked bit specifications in the AND mode or the satisfaction of any nonmasked bit specifications in either of the OR or OR-Priority Encoded Vector modes.

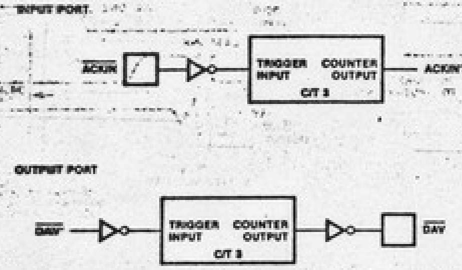


**Figure 9. Pulsed Handshake**

The pattern specified in the Pattern Definition register assumes that the data path is programmed to be noninverting. If an input bit in the data path is programmed to be inverting, the pattern detected is the opposite of the one specified. Output bits used in the pattern-match logic are internally sampled before the invert/noninvert logic.

**Bit Port Pattern-Recognition Operations.** During bit port operations, pattern-recognition may be performed on all bits, including those used as I/O for the counter/timers. The input to the pattern-recognition logic follows the value at the pins (through the invert/noninvert logic) in all cases except for simple inputs with 1's catchers. In this case, the output of the 1's catcher is used. When operating in the AND or OR mode, it is the transition from a no-match to a match state that causes the interrupt. In the "OR" mode, if a second match occurs before the first match goes away, it does not cause an interrupt. Since a match condition only lasts a short time when edges are specified, care must be taken to avoid losing a match condition. Bit ports specified in the OR-Priority Encoded Vector mode generate interrupts as long as any match state exists. A transition from a no-match to a match state is not required.

The pattern-recognition logic of bit ports operates in two basic modes: transparent and latched. When the Latch on Pattern Match (LPM) bit is set to 0 (Transparent mode), the interrupt indicates that a specified pattern has occurred, but a read of the Data register does not necessarily indicate the state of the port at the time the interrupt was generated. In the Latched mode (LPM = 1), the state of all the port inputs at the time the interrupt was generated is latched in the input register and held until IP is cleared. In all cases, the PMF indicates the state of the port at the time it is read.

If a match occurs while IP is already set, an error condition exists. If the Interrupt On Error bit (IOE) is 0, the match is ignored. However, if IOE is 1 after the first IP is cleared, it is automatically set to 1 along with the Interrupt Error (ERR) flag. Matches occurring while ERR is set are ignored. ERR is cleared when the corresponding IP is cleared.

When a pattern-match is present in the OR-Priority Encoded Vector mode, IP is set to 1. The IP cannot be cleared until a match is no longer present. If the interrupt vector is allowed to include status, the vector returned during Interrupt Acknowledge indicates the highest priority bit matching its specification at the time of the Acknowledge cycle. Bit 7 is the highest priority and bit 0 is the lowest. The bit initially causing the interrupt may not be the one indicated by the vector if a higher priority bit matches before the Acknowledge. Once the

## Functional Description (Continued)

Acknowledge cycle is initiated, the vector is frozen until the corresponding IP is cleared. Where inputs that cause interrupts might change before the interrupt is serviced, the 1's catcher can be used to hold the value. Because a no-match to match transition is not required, the source of the interrupt must be cleared before IP is cleared or else a second interrupt is generated. No error detection is performed in this mode, and the Interrupt On Error bit should be set to 0.

**Ports with Handshake Pattern-Recognition Operation.** In this mode, the handshake logic normally controls the setting of IP and, therefore, the generation of interrupt requests. The pattern-match logic controls the Pattern-Match Flag (PMF). The data is compared with the match pattern when it is shifted from the Buffer register to the Input register (input port) or when it is shifted from the Output register to the Buffer register (output port). The pattern match logic can override the handshake logic in certain situations. If the port is programmed to interrupt when two bytes of data are available to be read or written, but the first byte matches the specified pattern, the pattern-recognition logic sets IP and generates an interrupt. While PMF is set, IP cannot be cleared by reading or writing the data registers. IP must be cleared by command. The input register is not emptied while IP is set, nor is the output register filled until IP is cleared.

If the Interrupt on Match Only (IMO) bit is set, IP is set only when the data matches the pattern. This is useful in DMA-type application when interrupts are required only after a block of data is transferred.

**Counter/Timer Operation.** The three independent 16-bit counter/timers consist of a presettable 16-bit down counter, a 16-bit Time Constant register, a 16-bit Current Counter register, an 8-bit Mode Specification register, an 8-bit Command and Status register, and the associated control logic that links these registers.

| Function | $C/T_1$ | $C/T_2$ | $C/T_3$ |
|---|---|---|---|
| Counter/Timer Output | PB 4 | PB 0 | PC 0 |
| Counter Input | PB 5 | PB 1 | PC 1 |
| Trigger Input | PB 6 | PB 2 | PC 2 |
| Gate Input | PB 7 | PB 3 | PC 3 |

**Table 2. Counter/Timer External Access**

The flexibility of the counter/timers is enhanced by the provision of up to four lines per counter/timer (counter input, gate input, trigger input, and counter/timer output) for direct external control and status. Counter/Timer 1's external I/O lines are provided by the four most significant bits of Port B. Counter/Timer 2's are provided by the four least significant bits of Port B. Counter/Timer 3's external I/O lines are provided by the four bits of Port C. The utilization of these lines (Table 2) is programmable on a bit-by-bit basis via the Counter/Timer Mode Specification registers.

When external counter/timer I/O lines are to be used, the associated port lines must be vacant and programmed in the proper data direction. Lines used for counter/timer I/O have the same characteristics as simple input lines. They can be specified as inverting or noninverting; they can be read and used with the pattern-recognition logic. They can also include the 1's catcher input.

Counter/Timers 1 and 2 can be linked internally in three different ways. Counter/Timer 1's output (inverted) can be used as Counter/Timer 2's trigger, gate, or counter input. When linked, the counter/timers have the same capabilities as when used separately. The only restriction is that when Counter/Timer 1 drives Counter/Timer 2's count input, Counter/Timer 2 must be programmed with its external count input disabled.

There are three duty cycles available for the timer/counter output: pulse, one-shot, and square-wave. Figure 10 shows the counter/timer waveforms. When the Pulse mode

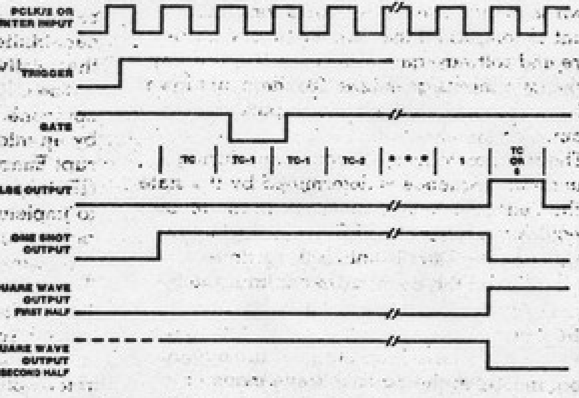


**Figure 10. Counter/Timer Waveforms**

**Functional Description (Continued)**

is specified, the output goes High for one clock cycle, beginning when the down-counter leaves the count of 1. In the One-Shot mode, the output goes High when the counter/timer is triggered and goes Low when the down-counter reaches 0. When the square-wave output duty cycle is specified, the counter/timer goes through two full sequences for each cycle. The initial trigger causes the down-counter to be loaded and the normal count-down sequence to begin. If a 1 count is detected on the down-counter's clocking edge, the output goes High and the time constant value is reloaded. On the clocking edge, when both the down-counter and the output are 1's, the output is pulled back Low.

The Continuous/Single Cycle (C/$\overline{SC}$) bit in the Mode Specification register controls operation of the down-counter when it reaches terminal count. If C/$\overline{SC}$ is 0 when a terminal count is reached, the countdown sequence stops. If the C/$\overline{SC}$ bit is 1 each time the count-down counter reaches 1, the next cycle causes the time constant value to be reloaded. The time constant value may be changed by the CPU, and on reload, the new time constant value is loaded.

Counter/timer operations require loading the time constant value in the Time Constant register and initiating the countdown sequence by loading the down-counter with the time constant value. The Time Constant register is accessed as two 8-bit registers. The registers are readable as well as writable, and the access order is irrelevant. A 0 in the Time Constant register specifies a time constant of 65,536. The down-counter is loaded in one of three ways: by writing a 1 to the Trigger Command Bit (TCB) of the Command and Status register, on the rising edge of the external trigger input, or, for Counter/Timer 2 only, on the rising edge of Counter/Timer 1's internal output if the counters are linked via the trigger input. The TCB is write-only, and read always returns 0.

Once the down-counter is loaded, the count-down sequence continues toward terminal count as long as all the counter/timers' hardware and software gate inputs are High. If any of the gate inputs goes Low (0), the countdown halts. It resumes when all gate inputs are 1 again.

The reaction to triggers occurring during a countdown sequence is determined by the state of the Retrigger Enable Bit (REB) in the Mode Specification register. If REB is 0, retriggers are ignored and the countdown continues normally. If REB is 1, each trigger causes the down-counter to be reloaded and the count-down sequence starts over again. If the output is programmed in the Square-Wave mode, retrigger causes the sequence to start over from the initial load of the time constant.

The rate at which the down-counter counts is determined by the mode of the counter/timer. In the Timer mode (the External Count Enable [ECE] bit is 0), the down-counter is clocked internally by a signal that is half the frequency of the PCLK input to the chip. In the Counter mode (ECE is 1), the down-counter is decremented on the rising edge of the counter/timer's counter input.

Each time the counter reaches terminal count, its Interrupt Pending (IP) bit is set to 1, and if interrupts are enabled (IE = 1), an interrupt is generated. If a terminal count occurs while IP is already set, an internal error flag is set. As soon as IP is cleared, it is forced to 1 along with the Interrupt Error (ERR) flag. Errors that occur after the internal flag is set are ignored.

The state of the down-counter can be determined in two ways: by reading the contents of the down-counter via the Current Count register or by testing the Count In Progress (CIP) status bit in the Command and Status register. The CIP status bit is set when the down-counter is loaded; it is reset when the down-counter reaches 0. The Current Count register is a 16-bit register, accessible as two 8-bit registers, which mirrors the contents of the down-counter. This register can be read anytime. However, reading the register is asynchronous to the counter's counting, and the value returned is valid only if the counter is stopped. The down-counter can be reliably read "on the fly" by the first writing of a 1 to the Read Counter Control (RCC) bit in the counter/timer's Command and Status register. This freezes the value in the Current Count register until a read of the least significant byte is performed.

**Interrupt Logic Operation.** The CIO has five potential sources of interrupts: the three counter/timers and Ports A and B. The priorities of these sources are fixed in the following order: Counter/Timer 3, Port A, Counter/Timer 2, Port B, and Counter/Timer 1. Since the counter/timers all have equal capabilities and Ports A and B have equal capabilities, there is no adverse impact from the relative priorities.

The CIO interrupt priority, relative to other components within the system, is determined by an interrupt daisy chain. Two pins, Interrupt Enable In (IEI) and Interrupt Enable Out (IEO), provide the input and output necessary to implement the daisy chain. When IEI is pulled Low by a higher priority device, the CIO cannot request an interrupt of the CPU. The following discussion assumes that the IEI line is High.

Each source of interrupt in the CIO contains three bits for the control and status of the interrupt logic: an Interrupt Pending (IP) status bit, an Interrupt Under Service (IUS)

**Functional Description** (Continued)

status bit, and an Interrupt Enable (IE) control bit. IP is set when an event requiring CPU intervention occurs. The setting of IP results in forcing the Interrupt ($\overline{INT}$) output Low, if the associated IE is 1.

The IUS status bit is set as a result of the Interrupt Acknowledge cycle by the CPU and is set only if its IP is of highest priority at the time the Interrupt Acknowledge commences. It can also be set directly by the CPU. Its primary function is to control the interrupt daisy chain. When set, it disables lower priority sources in the daisy chain, so that lower priority interrupt sources do not request servicing while higher priority devices are being serviced.

The IE bit provides the CPU with a means of masking off individual sources of interrupts. When IE is set to 1, interrupt is generated normally. When IE is set to 0, the IP bit is set when an event occurs that would normally require service; however, the $\overline{INT}$ output is not forced Low.

The Master Interrupt Enable (MIE) bit allows all sources of interrupts within the CIO to be disabled without having to individually set each IE to 0. If MIE is set to 0, all IPs are masked off and no interrupt can be requested or acknowledged. The Disable Lower Chain (DLC) bit is included to allow the CPU to modify the system daisy chain. When the DLC bit is set to 1, the CIO's IEO is forced Low, independent of the state of the CIO or its IEI input, and all lower priority devices' interrupts are disabled.

As part of the Interrupt Acknowledge cycle, the CIO is capable of responding with an 8-bit interrupt vector that specifies the source of the interrupt. The CIO contains three vector registers: one for Port A, one for Port B, and one shared by the three counter/timers. The vector output is inhibited by setting the No Vector (NV) control bit to 1. The vector output can be modified to include status information to pinpoint more precisely the cause of interrupt. Whether the vector includes status or not is controlled by a Vector Includes Status (VIS) control bit. Each base vector has its own VIS bit and is controlled independently. When MIE = 1, reading the base vector register always includes status, independent of the state of the VIS bit. In this way, all the information obtained by the vector, including status, can be obtained with one additional instruction when VIS is set to 0. When MIE = 0, reading the vector register returns the unmodified base vector so that it can be verified. Another register, the Current Vector register, allows use of the CIO in a polled environment. When read, the data returned is the same as the interrupt vector that would be output in an acknowledge, based on the highest priority IP set. If no unmasked IPs are set, the value $FF_H$ is returned. The Current Vector register is read-only.

**Programming**

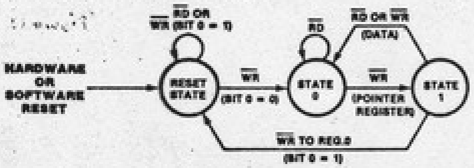
The data registers within the CIO are directly accessed by address lines $A_0$ and $A_1$ (Table 3). All other internal registers are accessed by the following two-step sequence, with the address lines specifying a control operation. First, write the address of the target register to an internal 6-bit Pointer Register; then read from or write to the target register. The Data registers can also be accessed by this method.

An internal state machine determines if accesses with $A_0$ and $A_1$ equalling 1 are to the Pointer Register or to an internal control register (Figure 11). Following any control read operation, the state machine is in State 0 (the next control access is to the Pointer Register). This can be used to force the state machine into a known state. Control reads in State 0 return the contents of the last register pointed to. Therefore, a register can be read continuously without writing to the Pointer. While the CIO is in State 1 (next control access is to the register pointed to), many internal operations are suspended—no IPs are set and internal status is frozen. Therefore, to minimize interrupt latency and to allow continuous status updates, the CIO should not be left in State 1.

The CIO is reset by forcing $\overline{RD}$ and $\overline{WR}$ Low simultaneously (normally an illegal condition) or by writing a 1 to the Reset bit. Reset disables all functions except a read from or write to the Reset bit; writes to all other bits are ignored, and all reads return $01_H$. In this state, all control bits are forced to 0 and may be programmed only after clearing the Reset bit (by writing a 0 to it).

| $A_1$ | $A_0$ | Register |
|---|---|---|
| 0 | 0 | Port C's Data Register |
| 0 | 1 | Port B's Data Register |
| 1 | 0 | Port A's Data Register |
| 1 | 1 | Control Registers |

**Table 3. Register Selection**



NOTE: State changes occur only when $A_0$ = $A_1$ = 1. No other accesses have effect.

**Figure 11. State Machine Operation**

## Registers

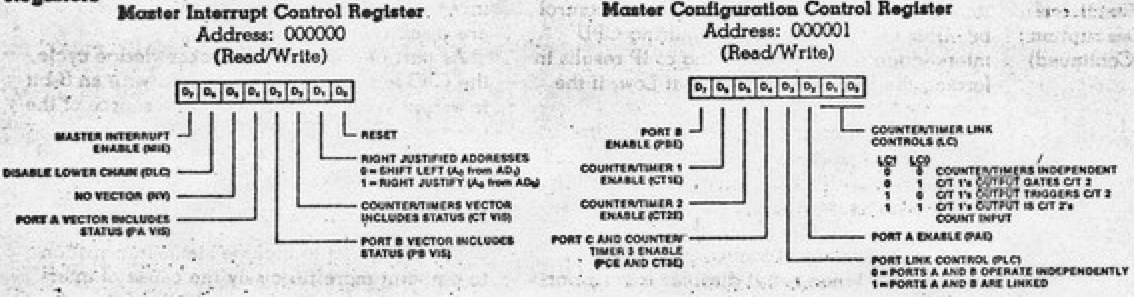

Master Interrupt Control Register
Address: 000000
(Read/Write)
D7 D6 D5 D4 D3 D2 D1 D0
MASTER INTERRUPT ENABLE (MIE)
DISABLE LOWER CHAIN (DLC)
NO VECTOR (NV)
PORT A VECTOR INCLUDES STATUS (PA VIS)
RESET
RIGHT JUSTIFIED ADDRESSES
0 = SHIFT LEFT (A0 from AD1)
1 = RIGHT JUSTIFY (A0 from AD0)
COUNTER/TIMERS VECTOR INCLUDES STATUS (CT VIS)
PORT B VECTOR INCLUDES STATUS (PB VIS)
Master Configuration Control Register
Address: 000001
(Read/Write)
D7 D6 D5 D4 D3 D2 D1 D0
PORT B ENABLE (PBE)
COUNTER/TIMER 1 ENABLE (CT1E)
COUNTER/TIMER 2 ENABLE (CT2E)
PORT C AND COUNTER/TIMER 3 ENABLE (PCE AND CT3E)
COUNTER/TIMER LINK CONTROLS (LC)
LC1 LC0
0 0 COUNTER/TIMERS INDEPENDENT
0 1 C/T 1's OUTPUT GATES C/T 2
1 0 C/T 1's OUTPUT TRIGGERS C/T 2
1 1 C/T 1's OUTPUT IS C/T 2's COUNT INPUT
PORT A ENABLE (PAE)
PORT LINK CONTROL (PLC)
0 = PORTS A AND B OPERATE INDEPENDENTLY
1 = PORTS A AND B ARE LINKED

**Figure 12. Master Control Registers**

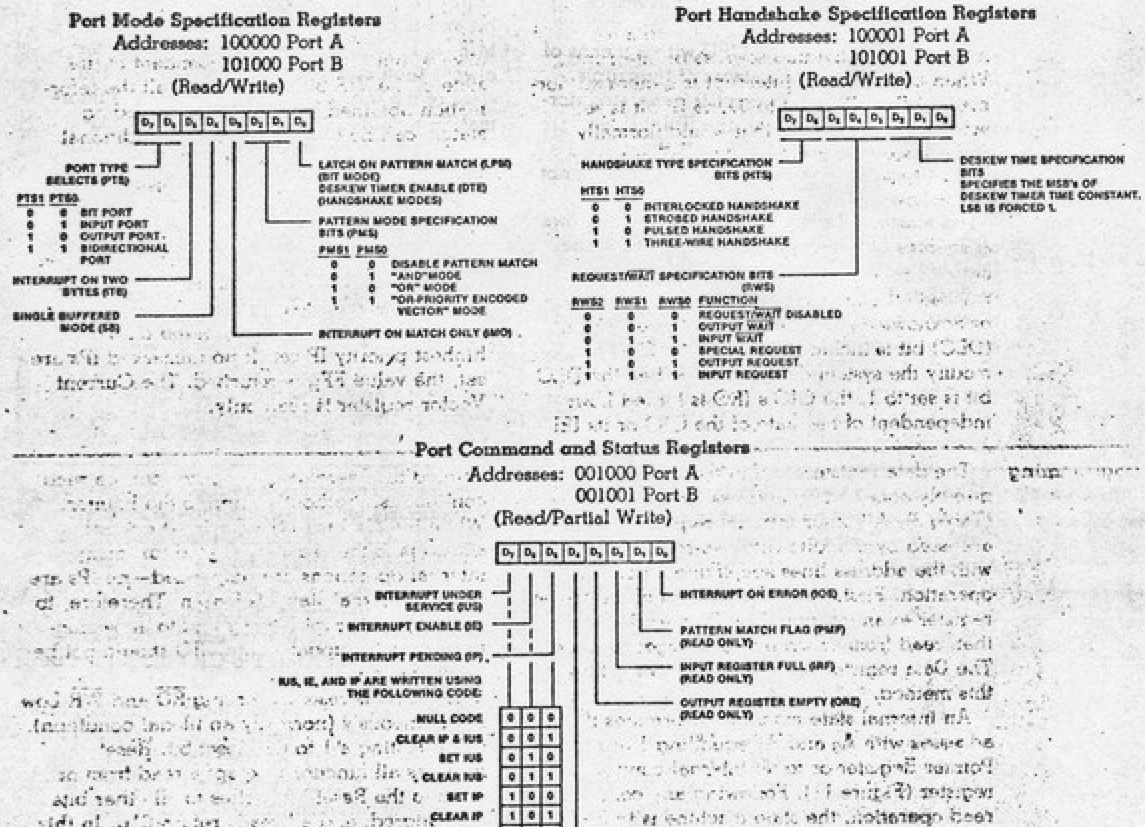
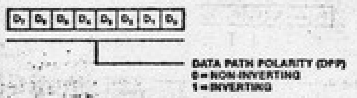

Port Mode Specification Registers
Addresses: 100000 Port A
101000 Port B
(Read/Write)
D7 D6 D5 D4 D3 D2 D1 D0
PORT TYPE SELECTS (PTS)
PTS1 PTS0
0 0 BIT PORT
0 1 INPUT PORT
1 0 OUTPUT PORT
1 1 BIDIRECTIONAL PORT
INTERRUPT ON TWO BYTES (ITB)
SINGLE BUFFERED MODE (SB)
LATCH ON PATTERN MATCH (LPM) (BIT MODE)
DESKEW TIMER ENABLE (DTE) (HANDSHAKE MODES)
PATTERN MODE SPECIFICATION BITS (PMS)
PMS1 PMS0
0 0 DISABLE PATTERN MATCH
0 1 "AND" MODE
1 0 "OR" MODE
1 1 "OR-PRIORITY ENCODED VECTOR" MODE
INTERRUPT ON MATCH ONLY (IMO)
Port Handshake Specification Registers
Addresses: 100001 Port A
101001 Port B
(Read/Write)
D7 D6 D5 D4 D3 D2 D1 D0
HANDSHAKE TYPE SPECIFICATION BITS (HTS)
HTS1 HTS0
0 0 INTERLOCKED HANDSHAKE
0 1 STROBED HANDSHAKE
1 0 PULSED HANDSHAKE
1 1 THREE-WIRE HANDSHAKE
REQUEST/WAIT SPECIFICATION BITS (RWS)
RWS2 RWS1 RWS0 FUNCTION
0 0 0 REQUEST/WAIT DISABLED
0 0 1 OUTPUT WAIT
0 1 1 INPUT WAIT
1 0 0 SPECIAL REQUEST
1 0 1 OUTPUT REQUEST
1 1 1 INPUT REQUEST
DESKEW TIME SPECIFICATION BITS
SPECIFIES THE MSB's OF DESKEW TIMER TIME CONSTANT. LSB IS FORCED 1.
Port Command and Status Registers
Addresses: 001000 Port A
001001 Port B
(Read/Partial Write)
D7 D6 D5 D4 D3 D2 D1 D0
INTERRUPT UNDER SERVICE (IUS)
INTERRUPT ENABLE (IE)
INTERRUPT PENDING (IP)
IUS, IE, AND IP ARE WRITTEN USING THE FOLLOWING CODE:
NULL CODE 0 0 0
CLEAR IP & IUS 0 0 1
SET IUS 0 1 0
CLEAR IUS 0 1 1
SET IP 1 0 0
CLEAR IP 1 0 1
SET IE 1 1 0
CLEAR IE 1 1 1
INTERRUPT ERROR (ERR) (READ ONLY)
INTERRUPT ON ERROR (IOE)
PATTERN MATCH FLAG (PMF) (READ ONLY)
INPUT REGISTER FULL (IRF) (READ ONLY)
OUTPUT REGISTER EMPTY (ORE) (READ ONLY)

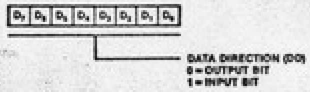
**Figure 13. Port Specifications Registers**

## Registers (Continued)

**Data Path Polarity Registers**
**Addresses:** 100010 Port A
101010 Port B
000101 Port C (4 LSBs only)
(Read/Write)



**Data Direction Registers**
**Addresses:** 100011 Port A
101011 Port B
000110 Port C (4 LSBs only)
(Read/Write)



**Special I/O Control Registers**
**Addresses:** 100100 Port A
101100 Port B
000111 Port C (4 LSBs only)
(Read/Write)

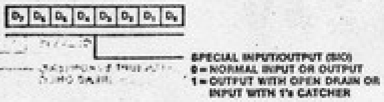

D7 D6 D5 D4 D3 D2 D1 D0
SPECIAL INPUT/OUTPUT (SIO)
0 = NORMAL INPUT OR OUTPUT
1 = OUTPUT WITH OPEN DRAIN OR
INPUT WITH 1's CATCHER

**Figure 14. Bit Path Definition Registers**

**Port Data Registers**
**Addresses:** 001101 Port A*
001110 Port B*
(Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

*These registers can be addressed directly.

**Port C Data Register**
Address: 001111*
(Read/Write)

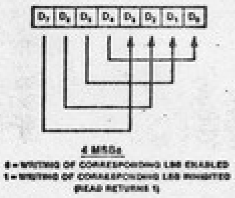
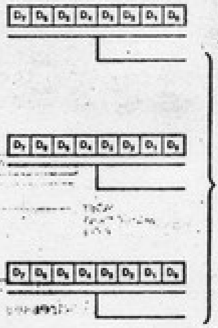

D7 D6 D5 D4 D3 D2 D1 D0
4 MSBs
0 = WRITING OF CORRESPONDING LSB ENABLED
1 = WRITING OF CORRESPONDING LSB INHIBITED
(READ RETURNS 1)

**Figure 15. Port Data Registers**

**Pattern Polarity Registers (PP)**
**Addresses:** 100101 Port A
101101 Port B
(Read/Write)

**Pattern Transition Registers (PT)**
**Addresses:** 100110 Port A
101110 Port B
(Read/Write)

**Pattern Mask Registers (PM)**
**Addresses:** 100111 Port A
101111 Port B
(Read/Write)



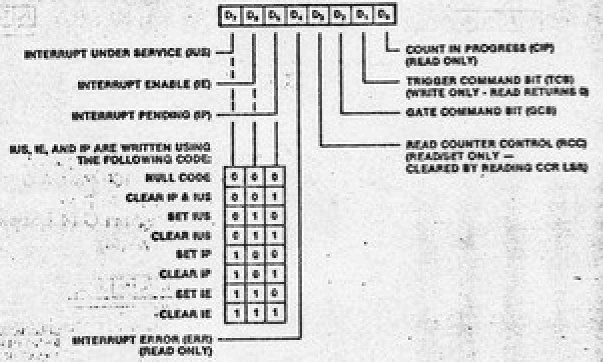
| PM | PT | PP | PATTERN SPECIFICATION |
|---|---|---|---|
| 0 | 0 | X | BIT MASKED OFF |
| 0 | 1 | X | ANY TRANSITION |
| 1 | 0 | 0 | ZERO |
| 1 | 0 | 1 | ONE |
| 1 | 1 | 0 | ONE TO ZERO TRANSITION (↘) |
| 1 | 1 | 1 | ZERO-TO-ONE TRANSITION (↗) |

**Figure 16. Pattern Definition Registers**

**Registers (Continued)**

### Counter/Timer Command and Status Registers

Addresses: 001010 Counter/Timer 1
001011 Counter/Timer 2
001100 Counter/Timer 3
(Read/Partial Write)

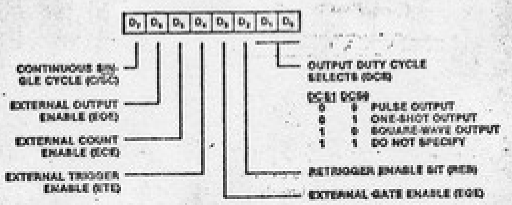

D7 D6 D5 D4 D3 D2 D1 D0
INTERRUPT UNDER SERVICE (IUS)
INTERRUPT ENABLE (IE)
INTERRUPT PENDING (IP)
INTERRUPT ERROR (ERR) (READ ONLY)
COUNT IN PROGRESS (CIP) (READ ONLY)
TRIGGER COMMAND BIT (TCB) (WRITE ONLY - READ RETURNS 0)
GATE COMMAND BIT (GCB)
READ COUNTER CONTROL (RCC) (READ/SET ONLY — CLEARED BY READING CCR LSB)

IUS, IE, AND IP ARE WRITTEN USING THE FOLLOWING CODE:

| | | | |
|---|---|---|---|
| NULL CODE | 0 | 0 | 0 |
| CLEAR IP & IUS | 0 | 0 | 1 |
| SET IUS | 0 | 1 | 0 |
| CLEAR IUS | 0 | 1 | 1 |
| SET IP | 1 | 0 | 0 |
| CLEAR IP | 1 | 0 | 1 |
| SET IE | 1 | 1 | 0 |
| CLEAR IE | 1 | 1 | 1 |

### Counter/Timer Mode Specification Registers

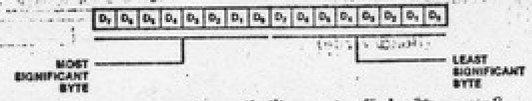
Addresses: 011100 Counter/Timer 1
011101 Counter/Timer 2
011110 Counter/Timer 3
(Read/Write)


D7 D6 D5 D4 D3 D2 D1 D0
CONTINUOUS SINGLE CYCLE (C/SC)
EXTERNAL OUTPUT ENABLE (EOE)
EXTERNAL COUNT ENABLE (ECE)
EXTERNAL TRIGGER ENABLE (ETE)
OUTPUT DUTY CYCLE SELECTS (DCS)
RETRIGGER ENABLE BIT (REB)
EXTERNAL GATE ENABLE (EGE)

| DCS1 | DCS0 | |
|---|---|---|
| 0 | 0 | PULSE OUTPUT |
| 0 | 1 | ONE-SHOT OUTPUT |
| 1 | 0 | SQUARE-WAVE OUTPUT |
| 1 | 1 | DO NOT SPECIFY |

### Counter/Timer Current Count Registers

Addresses: 010000 Counter/Timer 1's MSB
010001 Counter/Timer 1's LSB
010010 Counter/Timer 2's MSB
010011 Counter/Timer 2's LSB
010100 Counter/Timer 3's MSB
010101 Counter/Timer 3's LSB
(Read Only)



### Counter/Timer Time Constant Registers

Addresses: 010110 Counter/Timer 1's MSB
010111 Counter/Timer 1's LSB
011000 Counter/Timer 2's MSB
011001 Counter/Timer 2's LSB
011010 Counter/Timer 3's MSB
011011 Counter/Timer 3's LSB
(Read/Write)

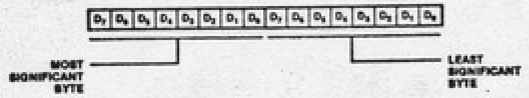


Figure 17. Counter/Timer Registers

## Registers (Continued)

**Interrupt Vector Register**
Addresses: 000010 Port A
000011 Port B
000100 Counter/Timers
(Read/Write)

**Current Vector Register**
Address: 011111
(Read only)

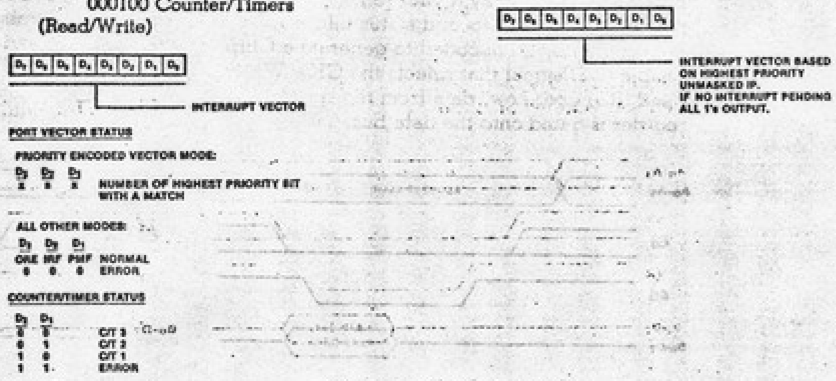


**Figure 18. Interrupt Vector Registers**

## Register Address Summary

**Main Control Registers**

| Address | Register Name |
|---|---|
| 000000 | Master Interrupt Control |
| 000001 | Master Configuration Control |
| 000010 | Port A's Interrupt Vector |
| 000011 | Port B's Interrupt Vector |
| 000100 | Counter/Timer's Interrupt Vector |
| 000101 | Port C's Data Path Polarity |
| 000110 | Port C's Data Direction |
| 000111 | Port C's Special I/O Control |

**Most Often Accessed Registers**

| Address | Register Name |
|---|---|
| 001000 | Port A's Command and Status |
| 001001 | Port B's Command and Status |
| 001010 | Counter/Timer 1's Command and Status |
| 001011 | Counter/Timer 2's Command and Status |
| 001100 | Counter/Timer 3's Command and Status |
| 001101 | Port A's Data (can be accessed directly) |
| 001110 | Port B's Data (can be accessed directly) |
| 001111 | Port C's Data (can be accessed directly) |

**Counter/Timer Related Registers**

| Address | Register Name |
|---|---|
| 010000 | Counter/Timer 1's Current Count-MSBs |
| 010001 | Counter/Timer 1's Current Count-LSBs |
| 010010 | Counter/Timer 2's Current Count-MSBs |
| 010011 | Counter/Timer 2's Current Count-LSBs |
| 010100 | Counter/Timer 3's Current Count-MSBs |
| 010101 | Counter/Timer 3's Current Count-LSBs |
| 010110 | Counter/Timer 1's Time Constant-MSBs |
| 010111 | Counter/Timer 1's Time Constant-LSBs |
| 011000 | Counter/Timer 2's Time Constant-MSBs |
| 011001 | Counter/Timer 2's Time Constant-LSBs |
| 011010 | Counter/Timer 3's Time Constant-MSBs |
| 011011 | Counter/Timer 3's Time Constant-LSBs |
| 011100 | Counter/Timer 1's Mode Specification |
| 011101 | Counter/Timer 2's Mode Specification |
| 011110 | Counter/Timer 3's Mode Specification |
| 011111 | Current Vector |

**Port A Specification Registers**

| Address | Register Name |
|---|---|
| 100000 | Port A's Mode Specification |
| 100001 | Port A's Handshake Specification |
| 100010 | Port A's Data Path Polarity |
| 100011 | Port A's Data Direction |
| 100100 | Port A's Special I/O Control |
| 100101 | Port A's Pattern Polarity |
| 100110 | Port A's Pattern Transition |
| 100111 | Port A's Pattern Mask |

**Port B Specification Registers**

| Address | Register Name |
|---|---|
| 101000 | Port B's Mode Specification |
| 101001 | Port B's Handshake Specification |
| 101010 | Port B's Data Path Polarity |
| 101011 | Port B's Data Direction |
| 101100 | Port B's Special I/O Control |
| 101101 | Port B's Pattern Polarity |
| 101110 | Port B's Pattern Transition |
| 101111 | Port B's Pattern Mask |

## Timing

**Read Cycle.** At the beginning of a read cycle, the CPU places an address on the address bus. Bits $A_0$ and $A_1$ specify a CIO register; the remaining address bits and status information are combined and decoded to generate a Chip Enable ($\overline{\text{CE}}$) signal that selects the CIO. When Read ($\overline{\text{RD}}$) goes Low, data from the specified register is gated onto the data bus.

**Write Cycle.** At the beginning of a write cycle, the CPU places an address on the data bus. Bits $A_0$ and $A_1$ specify a CIO register; the remaining address bits and status information are combined and decoded to generate a Chip Enable ($\overline{\text{CE}}$) signal that selects the CIO. When $\overline{\text{WR}}$ goes Low, data placed on the bus by the CPU is strobed into the specified CIO register.

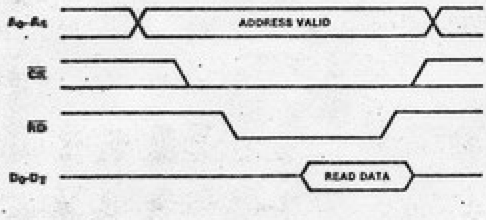


**Figure 19. Read Cycle Timing**

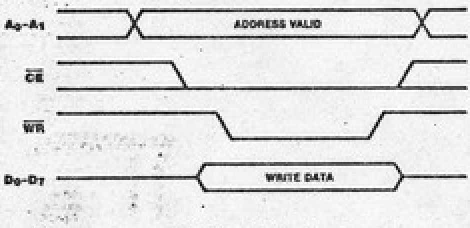


**Figure 20. Write Cycle Timing**

**Interrupt Acknowledge.** The CIO pulls its Interrupt Request ($\overline{\text{INT}}$) line Low, requesting interrupt service from the CPU, if an Interrupt Pending (IP) bit is set and interrupts are enabled. The CPU responds with an Interrupt Acknowledge cycle. When Interrupt Acknowledge ($\overline{\text{INTACK}}$) goes true and the IP is set, the CIO forces Interrupt Enable Out (IEO) Low, disabling all lower priority devices in the interrupt daisy chain. If the CIO is the highest priority device requesting service (IEI is High), it places its interrupt vector on the data bus and sets the Interrupt Under Service (IUS) bit when Read ($\overline{\text{RD}}$) goes Low.

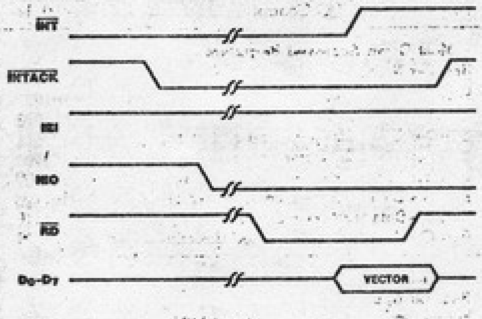


**Figure 21. Interrupt Acknowledge Timing**

### Absolute Maximum Ratings

Voltages on all inputs and outputs with respect to GND..........-0.3 V to +7.0 V

Operating Ambient Temperature ..................As Specified in Ordering Information

Storage Temperature........-65°C to +150°C

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

- $+4.75\ V \le V_{CC} \le +5.25\ V$
- GND = 0 V
- $T_A$ as specified in Ordering Information

All ac parameters assume a load capacitance of 50 pF max.

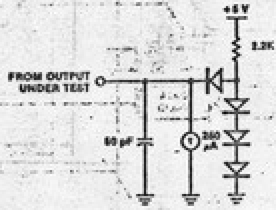


Figure 22. Standard Test Load

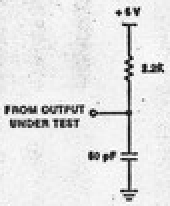


Figure 23. Open-Drain Test Load

### DC Characteristics

| Symbol | Parameter | Min | Max | Unit | Condition |
|---|---|---|---|---|---|
| $V_{IH}$ | Input High Voltage | 2.0 | $V_{CC}$+0.3 | V | |
| $V_{IL}$ | Input Low Voltage | -0.3 | 0.8 | V | |
| $V_{OH}$ | Output High Voltage | 2.4 | | V | $I_{OH} = -250\ \mu A$ |
| $V_{OL}$ | Output Low Voltage | | 0.4 | V | $I_{OL} = +2.0$ mA |
| | | | 0.5 | V | $I_{OL} = +3.2$ mA |
| $I_{IL}$ | Input Leakage | | ±10.0 | µA | $0.4 \le V_{IN} \le +2.4$ V |
| $I_{OL}$ | Output Leakage | | ±10.0 | µA | $0.4 \le V_{OUT} \le +2.4$ V |
| $I_{CC}$ | $V_{CC}$ Supply Current | | 200 | mA | |

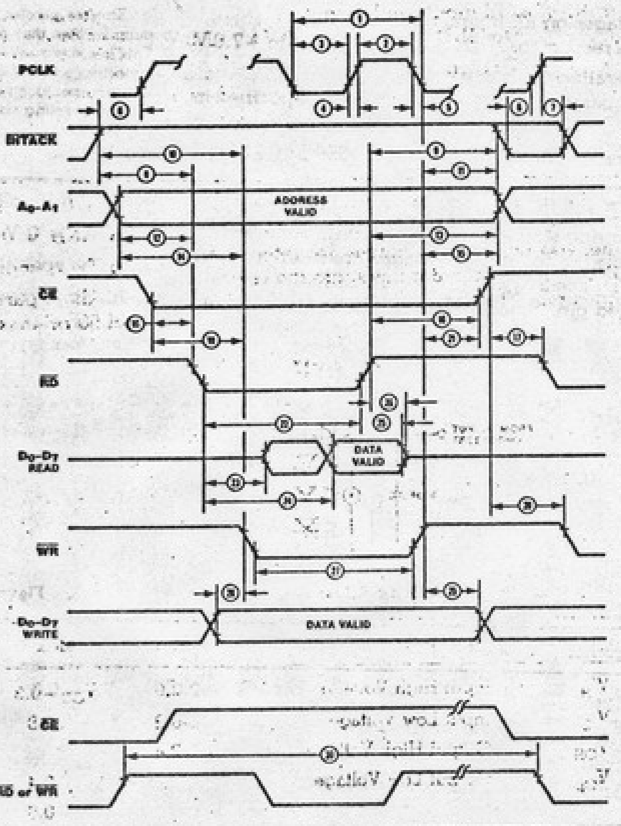
$V_{CC}$ = 5 V ± 5% unless otherwise specified, over specified temperature range.

### Capacitance

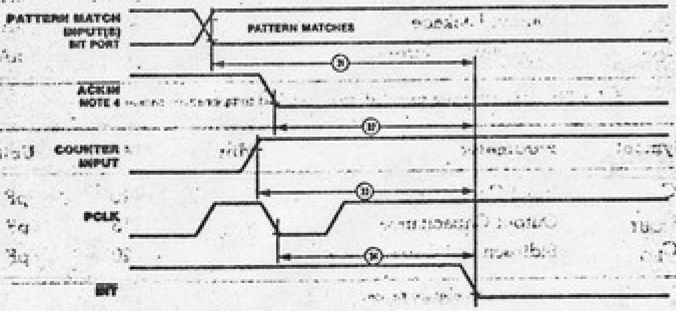
| Symbol | Parameter | Min | Max | Unit | Test Condition |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance | | 10 | pF | Unmeasured Pins Returned to Ground |
| $C_{OUT}$ | Output Capacitance | | 15 | pF | |
| $C_{I/O}$ | Bidirectional Capacitance | | 20 | pF | |

f = 1 MHz, over specified temperature range.

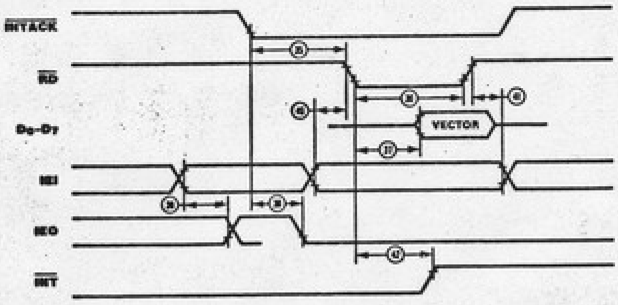
## CPU Interface Timing



## Interrupt Timing



## Interrupt Acknowledge Timing



| No. | Symbol | Parameter | 4 MHz |  | 6 MHz |  | Notes*† |
|---|---|---|---|---|---|---|---|
|  |  |  | Min | Max | Min | Max |  |
| 1 | TcPC | PCLK Cycle time | 250 | 4000 | 165 | 4000 |  |
| 2 | TwPCh | PCLK Width (High) | 105 | 2000 | 70 | 2000 |  |
| 3 | TwPCl | PCLK Width (Low) | 105 | 2000 | 70 | 2000 |  |
| 4 | TrPC | PCLK Rise Time |  | 20 |  | 10 |  |
| 5 | TfPC | PCLK Fall Time |  | 20 |  | 15 |  |
| 6 | TsIA(PC) | $\overline{\text{INTACK}}$ to PCLK ↑ Setup Time | 100 |  | 100 |  |  |
| 7 | ThIA(PC) | $\overline{\text{INTACK}}$ to PCLK ↑ Hold Time | 0 |  | 0 |  |  |
| 8 | TsIA(RD) | $\overline{\text{INTACK}}$ to $\overline{\text{RD}}$ ↓ Setup Time | 200 |  | 200 |  |  |
| 9 | ThIA(RD) | $\overline{\text{INTACK}}$ to $\overline{\text{RD}}$ ↑ Hold Time | 0 |  | 0 |  |  |
| 10 | TsIA(WR) | $\overline{\text{INTACK}}$ to $\overline{\text{WR}}$ ↓ Setup Time | 200 |  | 200 |  |  |
| 11 | ThIA(WR) | $\overline{\text{INTACK}}$ to $\overline{\text{WR}}$ ↑ Hold Time | 0 |  | 0 |  |  |
| 12 | TsA(RD) | Address to $\overline{\text{RD}}$ ↓ Setup Time | 80 |  | 80 |  |  |
| 13 | ThA(RD) | Address to $\overline{\text{RD}}$ ↑ Hold Time | 0 |  | 0 |  |  |
| 14 | TsA(WR) | Address to $\overline{\text{WR}}$ ↓ Setup Time | 80 |  | 80 |  |  |
| 15 | ThA(WR) | Address to $\overline{\text{WR}}$ ↑ Hold Time | 0 |  | 0 |  |  |
| 16 | TsCEl(RD) | $\overline{\text{CE}}$ Low to $\overline{\text{RD}}$ ↓ Setup Time | 0 |  | 0 |  | 1 |
| 17 | TsCEh(RD) | $\overline{\text{CE}}$ High to $\overline{\text{RD}}$ ↓ Setup Time | 100 |  | 70 |  | 1 |
| 18 | ThCE(RD) | $\overline{\text{CE}}$ to $\overline{\text{RD}}$ ↑ Hold Time | 0 |  | 0 |  | 1 |
| 19 | TsCEl(WR) | $\overline{\text{CE}}$ Low to $\overline{\text{WR}}$ ↓ Setup Time | 0 |  | 0 |  |  |
| 20 | TsCEh(WR) | $\overline{\text{CE}}$ High to $\overline{\text{WR}}$ ↓ Setup Time | 100 |  | 70 |  |  |
| 21 | ThCE(WR) | $\overline{\text{CE}}$ to $\overline{\text{WR}}$ ↑ Hold Time | 0 |  | 0 |  |  |
| 22 | TwRDl | $\overline{\text{RD}}$ Low Width | 390 |  | 250 |  | 1 |
| 23 | TdRD(DRA) | $\overline{\text{RD}}$ ↓ to Read Data Active Delay | 0 |  | 0 |  |  |
| 24 | TdRDf(DR) | $\overline{\text{RD}}$ ↓ to Read Data Valid Delay |  | 255 |  | 180 |  |
| 25 | TdRDr(DR) | $\overline{\text{RD}}$ ↑ to Read Data Not Valid Delay | 0 |  | 0 |  |  |
| 26 | TdRD(DRz) | $\overline{\text{RD}}$ ↑ to Read Data Float Delay |  | 70 |  | 45 | 2 |
| 27 | TwWRl | $\overline{\text{WR}}$ Low Width | 390 |  | 250 |  |  |
| 28 | TsDW(WR) | Write Data to $\overline{\text{WR}}$ ↓ Setup Time | 0 |  | 0 |  |  |
| 29 | ThDW(WR) | Write Data to $\overline{\text{WR}}$ ↑ Hold Time | 0 |  | 0 |  |  |
| 30 | Trc | Valid Access Recovery Time | 1000 |  | 650 |  | 3 |
| 31 | TdPM(INT) | Pattern Match to $\overline{\text{INT}}$ Delay (Bit Port) |  | 2+800 |  | 2 | 6 |
| 32 | TdACK(INT) | ACKIN to $\overline{\text{INT}}$ Delay (Port with Handshake) |  | 10+600 |  | 10 | 4,6 |
| 33 | TdCI(INT) | Counter Input to $\overline{\text{INT}}$ Delay (Counter Mode) |  | 2+700 |  | 2 | 6 |
| 34 | TdPC(INT) | PCLK to $\overline{\text{INT}}$ Delay (Timer Mode) |  | 3+700 |  | 3 | 6 |
| 35 | TsIA(RDA) | $\overline{\text{INTACK}}$ to $\overline{\text{RD}}$ ↓ (Acknowledge) Setup Time | 350 |  | 250 |  | 5 |
| 36 | TwRDA | $\overline{\text{RD}}$ (Acknowledge) Width | 350 |  | 250 |  |  |
| 37 | TdRDA(DR) | $\overline{\text{RD}}$ ↓ (Acknowledge) to Read Data Valid Delay |  | 250 |  | 180 |  |
| 38 | TdIA(IEO) | $\overline{\text{INTACK}}$ ↓ to IEO ↓ Delay |  | 350 |  | 250 | 5 |
| 39 | TdIEI(IEO) | IEI to IEO Delay |  | 150 |  | 100 | 5 |
| 40 | TsIEI(RDA) | IEI to $\overline{\text{RD}}$ ↓ (Acknowledge) Setup Time | 100 |  | 70 |  | 5 |
| 41 | ThIEI(RDA) | IEI to $\overline{\text{RD}}$ ↑ (Acknowledge) Hold Time | 100 |  | 70 |  |  |
| 42 | TdRDA(INT) | $\overline{\text{RD}}$ ↓ (Acknowledge) to $\overline{\text{INT}}$ ↑ Delay |  | 600 |  | 600 |  |

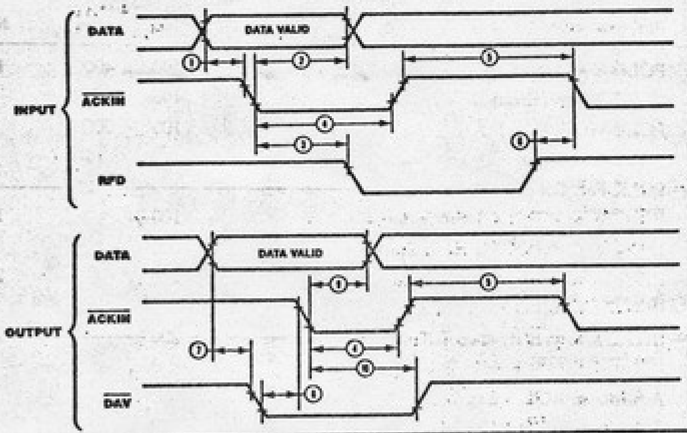
NOTES:

1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is measured to the time when the output has changed 0.5 V with minimum ac load and maximum dc load.
3. Trc is the specified number or 3 TcPC, whichever is longer.
4. The delay is from DAV ↓ for 3-Wire Input Handshake. The delay is from DAC ↑ for 3-Wire Output Handshake.
5. The parameters for the devices in any particular daisy chain must meet the following constraint: The delay from INTACK ↓ to RD ↓ must be greater than the sum of TdIA(IEO) for the highest priority peripheral, TsIEI(RDA) for the lowest priority peripheral, and TdIEI(IEO) for each peripheral separating them in the chain.
6. Units are equal to TcPC plus ns.

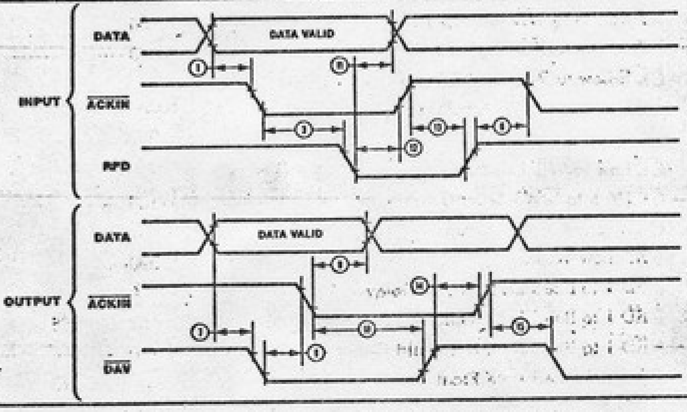
* Timings are preliminary and subject to change. All timing references assume 2.0 V for a logic "1", and 0.8 V for a logic "0".

† Units in nanoseconds (ns), except as noted.

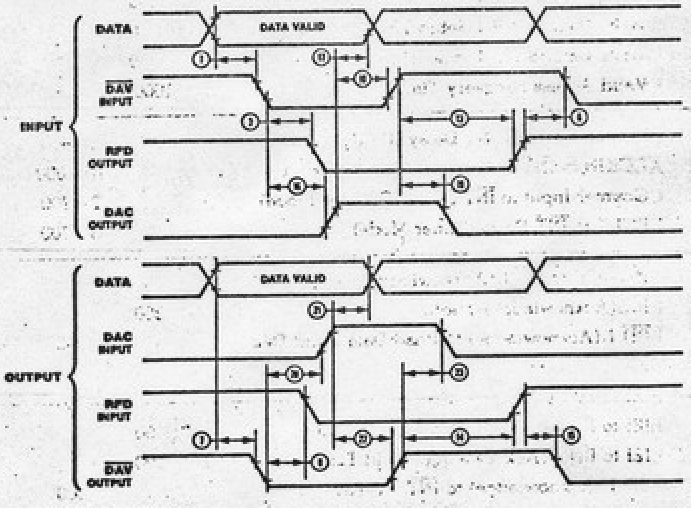
## Strobed Handshake



## Interlocked Handshake



## 3-Wire Handshake



| No. | Symbol | Parameter | 4 MHz<br>Min | 4 MHz<br>Max | 6 MHz<br>Min | 6 MHz<br>Max | Notes*† |
|---|---|---|---|---|---|---|---|
| 1 | TsDI(ACK) | Data Input to $\overline{\text{ACKIN}}$ ↓ Setup Time | 0 | | 0 | | |
| 2 | ThDI(ACK) | Data Input to $\overline{\text{ACKIN}}$ ↓ Hold Time—Strobed Handshake | | | | | |
| 3 | TdACKf(RFD) | $\overline{\text{ACKIN}}$ ↓ to RFD ↓ Delay | 0 | | 0 | | |
| 4 | TwACKl | $\overline{\text{ACKIN}}$ Low Width—Strobed Handshake | | | | | |
| 5 | TwACKh | $\overline{\text{ACKIN}}$ High Width—Strobed Handshake | | | | | |
| 6 | TdRFDr(ACK) | RFD ↑ to $\overline{\text{ACKIN}}$ ↓ Delay | 0 | | 0 | | |
| 7 | TsDO(DAV) | Data Out to $\overline{\text{DAV}}$ ↓ Setup Time | 25 | | 20 | | 1 |
| 8 | TdDAVf(ACK) | $\overline{\text{DAV}}$ ↓ to $\overline{\text{ACKIN}}$ ↓ Delay | 0 | | 0 | | |
| 9 | ThDO(ACK) | Data Out to $\overline{\text{ACKIN}}$ ↓ Hold Time | 2 | | 2 | | 2 |
| 10 | TdACK(DAV) | $\overline{\text{ACKIN}}$ ↓ to $\overline{\text{DAV}}$ ↑ Delay | 2 | | 2 | | 2 |
| 11 | THDI(RFD) | Data Input to RFD ↓ Hold Time—Interlocked Handshake | | | | | |
| 12 | TdRFDf(ACK) | RFD ↓ to $\overline{\text{ACKIN}}$ ↑ Delay Interlocked Handshake | 0 | | 0 | | |
| 13 | TdACKr(RFD) | $\overline{\text{ACKIN}}$ ↑ ($\overline{\text{DAV}}$ ↑) to RFD ↑ Delay—Interlocked and 3-Wire Handwshake | 0 | | 0 | | |
| 14 | TdDAVr(ACK) | $\overline{\text{DAV}}$ ↑ to $\overline{\text{ACKIN}}$ ↑ (RFD ↑)—Interlocked and 3-Wire Handshake | 0 | | 0 | | |
| 15 | TdACK(DAV) | $\overline{\text{ACKIN}}$ ↑ (RFD ↑) to $\overline{\text{DAV}}$ ↓ Delay—Interlocked and 3-Wire Handshake | 0 | | 0 | | |
| 16 | TdDAVIf(DAC) | $\overline{\text{DAV}}$ ↓ to DAC ↑ Delay—Input 3-Wire Handshake | 0 | | 0 | | |
| 17 | ThDI(DAC) | Data Input to DAC ↑ Hold Time—3-Wire Handshake | 0 | | 0 | | |
| 18 | TdDACOr(DAV) | DAC ↑ to $\overline{\text{DAV}}$ ↑ Delay—Input 3-Wire Handshake | 0 | | 0 | | |
| 19 | TdDAVIr(DAC) | $\overline{\text{DAV}}$ ↑ to DAC ↓ Delay—Input 3-Wire Handshake | 0 | | 0 | | |
| 20 | TdDAVOf(DAC) | $\overline{\text{DAV}}$ ↓ to DAC ↑ Delay—Output 3-Wire Handshake | 0 | | 0 | | |
| 21 | ThDO(DAC) | Data Output to DAC ↑ Hold Time—3-Wire Handshake | 2 | | 2 | | 2 |
| 22 | TdDACIr(DAV) | DAC ↑ to $\overline{\text{DAV}}$ ↑ Delay—Output 3-Wire Handshake | 2 | | 2 | | 2 |
| 23 | TdDAVOr(DAC) | $\overline{\text{DAV}}$ ↑ to DAC ↓ Delay—Output 3-Wire Handshake | 0 | | 0 | | |

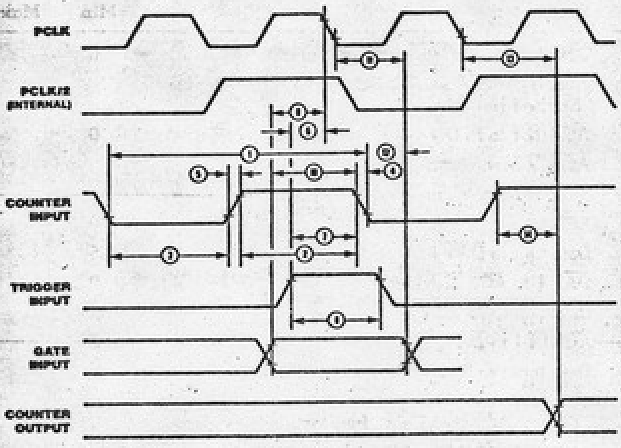
NOTES:

1. This time can be extended through the use of deskew timers.
2. Units equal to TcPC.

* Timings are preliminary and subject to change. All timing references assume 2.0 V for a logic "1" and 0.8 V for a logic "0".

† Units in nanoseconds (ns), except as noted.

## Counter/ Timer Timing

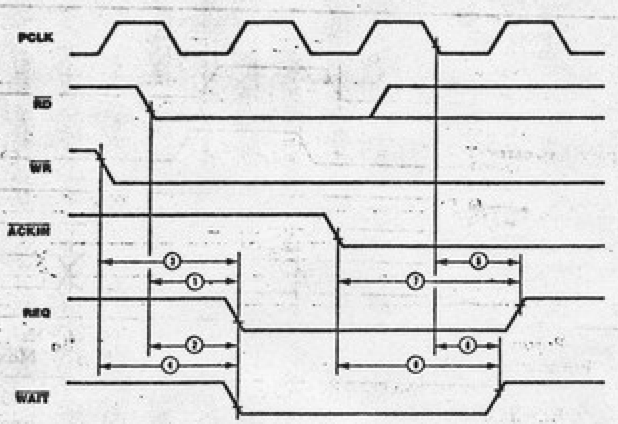


| No. | Symbol | Parameter | 4 MHz Min | 4 MHz Max | 6 MHz Min | 6 MHz Max | Notes*† |
|---|---|---|---|---|---|---|---|
| 1 | TcCI | Counter Input Cycle Time | 500 | | 330 | | |
| 2 | TCIh | Counter Input High Width | 230 | | 150 | | |
| 3 | TWCIl | Counter Input Low Width | 230 | | 150 | | |
| 4 | TfCI | Counter Input Fall Time | | 20 | | 15 | |
| 5 | TrCI | Counter Input Rise Time | | 20 | | 15 | |
| 6 | TsTI(PC) | Trigger Input to PCLK ↓ Setup Time (Timer Mode) | | | | | 1 |
| 7 | TsTI(CI) | Trigger Input to Counter Input ↓ Setup Time (Counter Mode) | | | | | 1 |
| 8 | TwTI | Trigger Input Pulse Width (High or Low) | | | | | |
| 9 | TsGI(PC) | Gate Input to PCLK ↓ Setup Time (Timer Mode) | | | | | 1 |
| 10 | TsGI(CI) | Gate Input to Counter Input ↓ Setup Time (Countег Mode) | | | | | 1 |
| 11 | ThGI(PC) | Gate Input to PCLK ↓ Hold Time (Timer Mode) | | | | | 1 |
| 12 | ThGI(CI) | Gate Input to Counter Input ↓ Hold Time (Counter Mode) | | | | | 1 |
| 13 | TdPC(CO) | PCLK to Counter Output Delay (Timer Mode) | | | | | |
| 14 | TdCI(CO) | Counter Input to Counter Output Delay (Counter Mode) | | | | | |

NOTES:
1. These parameters must be met to guarantee trigger or gate are valid for the next counter/timer cycle.

* Timings are preliminary and subject to change. All timing references assume 2.0 V for a logic "1" and 0.8 V for a logic "0".
† Units in nanoseconds (ns).

## REQUEST/ $\overline{\text{WAIT}}$ Timing

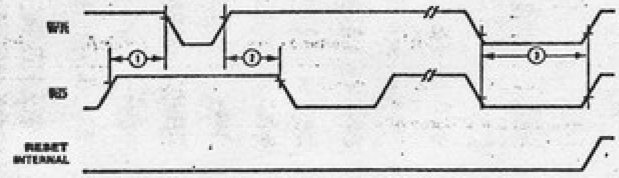


|  |  |  | 4 MHz |  | 6 MHz |  |  |
|---|---|---|---|---|---|---|---|
| No. | Symbol | Parameter | Min | Max | Min | Max | Notes*† |
| 1 | TdRD(REQ) | $\overline{\text{RD}}$ ↓ to REQ ↓ Delay |  | 500 |  |  |  |
| 2 | TdRD(WAIT) | $\overline{\text{RD}}$ ↓ to $\overline{\text{WAIT}}$ ↓ Delay |  | 500 |  |  |  |
| 3 | TdWR(REQ) | $\overline{\text{WR}}$ ↓ to REQ ↓ Delay |  | 500 |  |  |  |
| 4 | TdWR(WAIT) | $\overline{\text{WR}}$ ↓ to $\overline{\text{WAIT}}$ ↓ Delay |  | 500 |  |  |  |
| 5 | TdPC(REQ) | PCLK ↓ to REQ ↑ Delay |  | 300 |  |  |  |
| 6 | TdPC(WAIT) | PCLK ↓ to $\overline{\text{WAIT}}$ ↑ Delay |  | 300 |  |  |  |
| 7 | TdACK(REQ) | $\overline{\text{ACKIN}}$ ↓ to REQ ↑ Delay |  | 8 + 100 |  |  | 1,2 |
| 8 | TdACK(WAIT) | $\overline{\text{ACKIN}}$ ↓ to $\overline{\text{WAIT}}$ ↑ Delay |  | 10 + 600 |  |  | 1,2 |

NOTES:
1. The delay is fronm DAV ↓ for 3-Wire Input Handshake. The delay is from DAC ↑ for 3-Wire Output Handshake.
2. Units equal to TcPC + ns.

* Timings are preliminary and subject to change. All timing references assume 2.0 V for a logic "1" and 0.8 V for a logic "0".
† Units in nanoseconds (ns), except as noted.

## Reset Timing

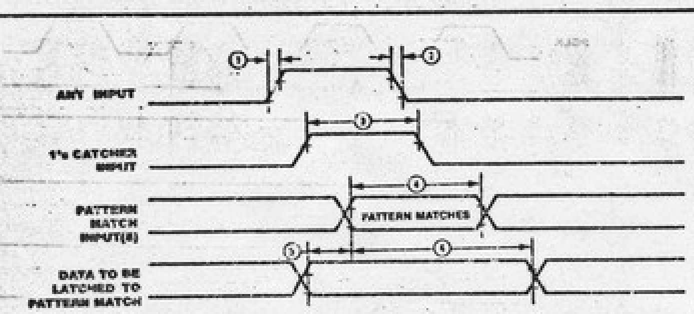


|  |  |  | 4 MHz |  | 6 MHz |  |  |
|---|---|---|---|---|---|---|---|
| No. | Symbol | Parameter | Min | Max | Min | Max | Notes*† |
| 1 | TdRD(WR) | Delay from $\overline{\text{RD}}$ ↑ to $\overline{\text{WR}}$ ↓ for No Reset | 50 |  | 50 |  |  |
| 2 | TdWR(RD) | Delay from $\overline{\text{WR}}$ ↑ to $\overline{\text{RD}}$ ↓ for No Reset | 50 |  | 50 |  |  |
| 3 | TwRES | Minimum Width of $\overline{\text{RD}}$ and $\overline{\text{WR}}$ both Low for Reset | 250 |  | 250 |  |  |

* Timings are preliminary and subject to change. All timing references assume 2.0 V for a logic "1" and 0.8 V for a logic "0".

† Unites in nanoseconds (ns).

## Miscellaneous Port Timing



| No. | Symbol | Parameter | 4 MHz Min | 4 MHz Max | 6 MHz Min | 6 MHz Max | Notes*† |
|---|---|---|---|---|---|---|---|
| 1 | TrI | Any Input Rise Time | | 100 | | 100 | |
| 2 | TfI | Any Input Fall Time | | 100 | | 100 | |
| 3 | Tw1's | 1's Catcher High Width | 250 | | 170 | | 1 |
| 4 | TwPM | Pattern Match Input Valid (Bit Port) | 750 | | 500 | | |
| 5 | TsPMD | Data Latched on Pattern Match Setup Time (Bit Port) | 0 | | 0 | | |
| 6 | ThPMD | Data Latched on Pattern Match Hold Time (Bit Port) | 1000 | | 650 | | |

NOTES:
1. If the input is programmed inverting, a Low-going pulse of the same width will be detected.

* Timings are preliminary and subject to change. All timing references assume 2.0 V for a logic "1" and 0.8 V for a logic "0".
† Units in nanoseconds (ns).

## Ordering Information

| Product Number | Package/Temp | Speed | Description | Product Number | Package/Temp | Speed | Description |
|---|---|---|---|---|---|---|---|
| Z8536 | CE | 4.0 MHz | CIO (40-pin) | Z8536A | CE | 6.0 MHz | CIO (40-pin) |
| Z8536 | CM | 4.0 MHz | Same as above | Z8536A | CM | 6.0 MHz | Same as above |
| Z8536 | CMB | 4.0 MHz | Same as above | Z8536A | CMB | 6.0 MHz | Same as above |
| Z8536 | CS | 4.0 MHz | Same as above | Z8536A | CS | 6.0 MHz | Same as above |
| Z8536 | DE | 4.0 MHz | Same as above | Z8536A | DE | 6.0 MHz | Same as above |
| Z8536 | DS | 4.0 MHz | Same as above | Z8536A | DS | 6.0 MHz | Same as above |
| Z8536 | PE | 4.0 MHz | Same as above | Z8536A | PE | 6.0 MHz | Same as above |
| Z8536 | PS | 4.0 MHz | Same as above | Z8536A | PS | 6.0 MHz | Same as above |

NOTES: C = Ceramic, D = Cerdip, P = Plastic; E = -40°C to +85°C, M = -55°C to 125°C, MB = -55°C to 125°C with MIL-STD-883 with Class B processing, S = 0°C to +70°C.

Part No.: 32 958 044
Related doc: 4

MOTOROLA

# MC146818

## Advance Information

CMOS

(HIGH-PERFORMANCE
SILICON-GATE COMPLEMENTARY MOS)

REAL-TIME CLOCK
PLUS RAM

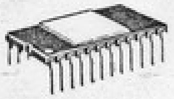
### REAL-TIME CLOCK PLUS RAM (RTC)

The MC146818 Real-Time Clock plus RAM is a peripheral device which includes the unique MOTEL concept for use with various microprocessors, microcomputers, and larger computers. This part combines three unique features: a complete time-of-day clock with alarm and one hundred year calendar, a programmable periodic interrupt and square-wave generator, and 50 bytes of low-power static RAM. The MC146818 uses high-speed CMOS technology to interface with 1 MHz processor buses, while consuming very little power.

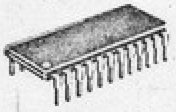
The Real-Time Clock plus RAM has two distinct uses. First, it is designed as a battery powered CMOS part (in an otherwise NMOS/TTL system) including all the common battery backed-up functions such as RAM, time, and calendar. Secondly, the MC146818 may be used with a CMOS microprocessor to relieve the software of the timekeeping workload and to extend the available RAM of an MPU such as the MC146805E2.

- Low-Power, High-Speed, High-Density CMOS
- Internal Time Base and Oscillator
- Counts Seconds, Minutes, and Hours of the Day
- Counts Days of the Week, Date, Month, and Year
- 3 V to 6 V Operation
- Time Base Input Options: 4.194304 MHz, 1.048576 MHz, or 32.768 kHz
- Time Base Oscillator for Parallel Resonant Crystals
- 40 to 200 μW Typical Operating Power at Low Frequency Time Base
- 4.0 to 20 mW Typical Operating Power at High Frequency Time Base
- Binary or BCD Representation of Time, Calendar, and Alarm
- 12- or 24-Hour Clock with AM and PM in 12-Hour Mode
- Daylight Savings Time Option
- Automatic End of Month Recognition
- Automatic Leap Year Compensation
- Microprocessor Bus Compatible
- MOTEL Circuit for Bus Universality
- Multiplexed Bus for Pin Efficiency
- Interfaced with Software as 64 RAM Locations
- 14 Bytes of Clock and Control Registers
- 50 Bytes of General Purpose RAM
- Status Bit Indicates Data Integrity
- Bus Compatible Interrupt Signals ($\overline{IRQ}$)
- Three Interrupts are Separately Software Maskable and Testable
  Time-of-Day Alarm, Once-per-Second to Once-per-Day
  Periodic Rates from 30.5 μs to 500 ms
  End-of-Clock Update Cycle
- Programmable Square-Wave Output Signal
- Clock Output May Be Used as Microprocessor Clock Input
  At Time Base Frequency ÷1 or ÷4
- 24-Pin Dual-In-Line Package
- Chip Carrier Also Available

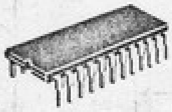
L SUFFIX
CERAMIC PACKAGE
CASE 716



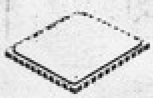
P SUFFIX
PLASTIC PACKAGE
CASE 709



S SUFFIX
CERDIP PACKAGE
CASE 623

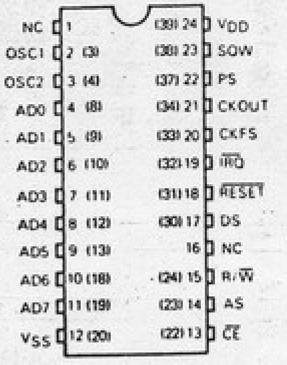


Z SUFFIX
CHIP CARRIER
CASE 761



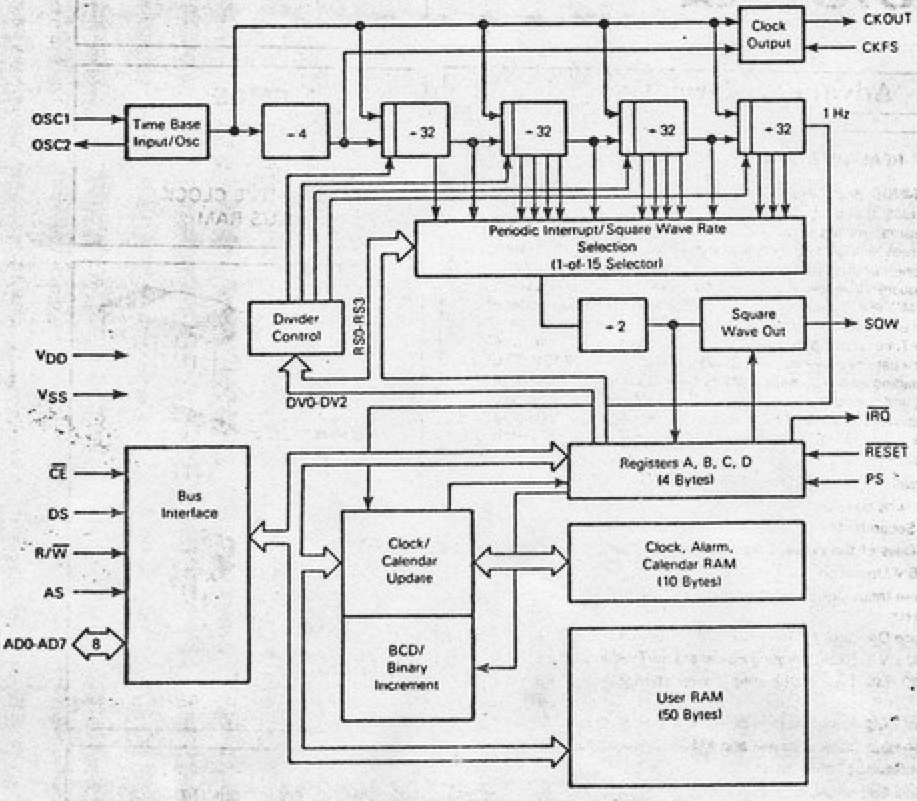
3

PIN ASSIGNMENT



Pin numbers in parentheses represent equivalent Z suffix chip carrier pins. Pins that have not been designated for the chip carrier are not connected.

FIGURE 1 — BLOCK DIAGRAM



## MAXIMUM RATINGS (Voltages referenced to $V_{SS}$)

| Ratings | Symbol | Value | Unit |
|---|---|---|---|
| Supply Voltage | $V_{DD}$ | −0.3 to +8.0 | V |
| All Input Voltages Except OSC1 | $V_{in}$ | $V_{SS}-0.5$ to $V_{DD}+0.5$ | V |
| Current Drain per Pin Excluding $V_{DD}$ and $V_{SS}$ | I | 10 | mA |
| Operating Temperature Range<br>MC146818<br>MC146818C ($V_{DD}$ = 3.0 to 5.5 V operation) | $T_A$ | $T_L$ to $T_H$<br>0 to 70<br>−40 to 85 | °C |
| Storage Temperature Range | $T_{stg}$ | −55 to +150 | °C |

## THERMAL CHARACTERISTICS

| Characteristic | Symbol | Value | Unit |
|---|---|---|---|
| Thermal Resistance<br>Plastic<br>Cerdip<br>Ceramic | $\theta_{JA}$ | <br>120<br>65<br>50 | °C/W |

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that $V_{in}$ and $V_{out}$ be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either $V_{SS}$ or $V_{DD}$).

**DC ELECTRICAL CHARACTERISTICS** ($V_{DD}$ = 3 Vdc, $V_{SS}$ = 0 Vdc, $T_A = T_L$ to $T_H$ unless otherwise noted)

| Characteristics | | Symbol | Min | Max | Unit |
|---|---|---|---|---|---|
| Frequency of Operation | | $f_{osc}$ | 32.768 | 32.768 | kHz |
| Output Voltage | | $V_{OL}$ | – | 0.1 | V |
| $I_{Load} < 10$ µA | | $V_{OH}$ | $V_{DD} - 0.1$ | – | |
| $I_{DD}$ – Bus Idle<br>CKOUT = $f_{osc}$, $C_L$ = 15 pF; SQW Disabled, $\overline{CE} = V_{DD} - 0.2$, $C_L$ (OSC2) = 10 pF<br>$f_{osc}$ = 32.768 kHz | | $I_{DD3}$ | – | 50 | µA |
| $I_{DD}$ – Quiescent<br>$f_{osc}$ = DC; OSC1 = DC;<br>All Other Inputs = $V_{DD} - 0.2$ V;<br>No Clock | | $I_{DD4}$ | – | 50 | µA |
| Output High Voltage<br>$(I_{LLoad} = -0.25$ mA, All Outputs) | | $V_{OH}$ | 2.7 | – | V |
| Output Low Voltage<br>$(I_{Load} = 0.25$ mA, All Outputs) | | $V_{OL}$ | – | 0.3 | V |
| Input High Voltage | AD0-AD7, DS, AS, $R/\overline{W}$, $\overline{CE}$,<br>$\overline{RESET}$, CKFS, PS, OSC1 | $V_{IH}$ | 2.1<br>2.5 | $V_{DD}$<br>$V_{DD}$ | V |
| Input Low Voltage (All Inputs) | | $V_{IL}$ | $V_{SS}$ | 0.5 | V |
| Input Current | All Inputs | $I_{in}$ | – | ± 1 | µA |
| Three-State Leakage | $\overline{IRQ}$, AD0-AD7 | $I_{TSL}$ | – | ± 10 | µA |

**DC ELECTRICAL CHARACTERISTICS** ($V_{DD}$ = 5 Vdc ± 10%, $V_{SS}$ = 0 Vdc, $T_A = T_L$ to $T_H$ unless otherwise noted)

| Characteristics | | Symbol | Min | Max | Unit |
|---|---|---|---|---|---|
| Frequency of Operation | | $f_{osc}$ | 32.768 | 4194.304 | kHz |
| Output Voltage | | $V_{OL}$ | – | 0.1 | V |
| $I_{Load} < 10$ µA | | $V_{OH}$ | $V_{DD} - 0.1$ | – | |
| $I_{DD}$ – Bus Idle (External Clock)<br>CKOUT = $f_{osc}$, $C_L$ = 15 pF; SQW Disabled, $\overline{CE} = V_{DD} - 0.2$; $C_L$ (OSC2) = 10 pF<br>$f_{osc}$ = 4.194304 MHz<br>$f_{osc}$ = 1.048516 MHz<br>$f_{osc}$ = 32.768 kHz | | $I_{DD1}$<br>$I_{DD2}$<br>$I_{DD3}$ | –<br>–<br>– | 3<br>800<br>50 | mA<br>µA<br>µA |
| $I_{DD}$ – Quiescent<br>$f_{osc}$ = DC; OSC1 = DC;<br>All Other Inputs = $V_{DD} - 0.2$ V;<br>No Clock | | $I_{DD4}$ | – | 50 | µA |
| Output High Voltage<br>$(I_{Load} = -1.6$ mA, AD0-AD7, CKOUT)<br>$(I_{Load} = -1.0$ mA, SQW) | | $V_{OH}$ | 4.1 | – | V |
| Output Low Voltage<br>$(I_{Load} = 1.6$ mA, AD0-AD7, CKOUT)<br>$(I_{Load} = 1.0$ mA, $\overline{IRQ}$ and SQW) | | $V_{OL}$ | – | 0.4 | V |
| Input High Voltage | CKFS, AD0-AD7, DS, AS, $R/\overline{W}$, $\overline{CE}$, PS<br>$\overline{RESET}$<br>OSC1 | $V_{IH}$ | $V_{DD} - 2.0$<br>$V_{DD} - 0.8$<br>$V_{DD} - 1.0$ | $V_{DD}$<br>$V_{DD}$<br>$V_{DD}$ | V |
| Input Low Voltage | AD0-AD7, DS, AS, $R/\overline{W}$, $\overline{CE}$<br>CKFS, PS, $\overline{RESET}$<br>OSC1 | $V_{IL}$ | $V_{SS}$<br>$V_{SS}$<br>$V_{SS}$ | 0.8<br>0.8<br>0.8 | V |
| Input Current | All Inputs | $I_{in}$ | – | ± 1 | µA |
| Three-State Leakage | $\overline{IRQ}$, AD0-AD7 | $I_{TSL}$ | – | ± 10 | µA |

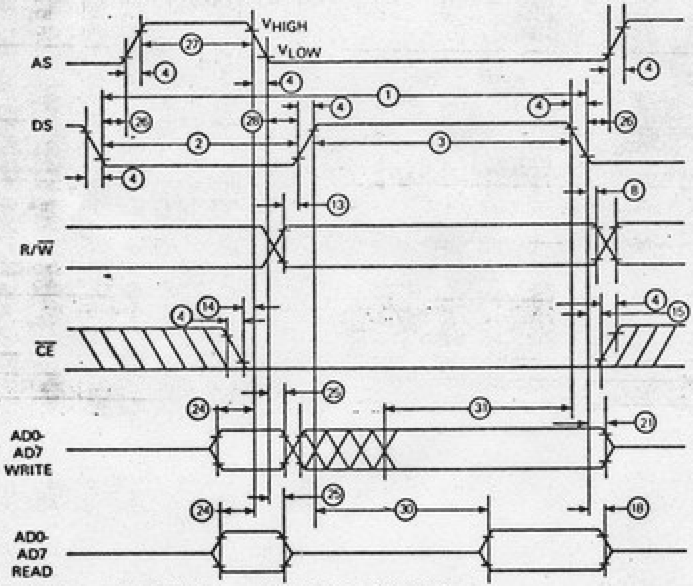
## BUS TIMING

| Ident. Number | Characteristics | Symbol | $V_{DD}$ = 3.0 V<br>50 pF Load |  | $V_{DD}$ = 5.0 V<br>± 10%<br>2 TTL and<br>130 pF Load |  | Unit |
|---|---|---|---|---|---|---|---|
|  |  |  | Min | Max | Min | Max |  |
| 1 | Cycle Time | $t_{cyc}$ | 5000 | – | 953 | dc | ns |
| 2 | Pulse Width, DS/E Low or $\overline{RD}$/$\overline{WR}$ High | $PW_{EL}$ | 1000 | – | 300 | – | ns |
| 3 | Pulse Width, DS/E High or $\overline{RD}$/$\overline{WR}$ Low | $PW_{EH}$ | 1500 | – | 325 | – | ns |
| 4 | Input Rise and Fall Time | $t_r$, $t_f$ | – | 100 | – | 30 | ns |
| 8 | R/$\overline{W}$ Hold Time | $t_{RWH}$ | 10 | – | 10 | – | ns |
| 13 | R/$\overline{W}$ Setup Time Before DS/E | $t_{RWS}$ | 200 | – | 80 | – | ns |
| 14 | Chip Enable Setup Time Before AS/ALE Fall | $t_{CS}$ | 200 | * | 55 | * | ns |
| 15 | Chip Enable Hold Time | $t_{CH}$ | 10 | – | 0 | – | ns |
| 18 | Read Data Hold Time | $t_{DHR}$ | 10 | 1000 | 10 | 100 | ns |
| 21 | Write Data Hold Time | $t_{DHW}$ | 100 | – | 0 | – | ns |
| 24 | Muxed Address Valid Time to AS/ALE Fall | $t_{ASL}$ | 200 | – | 50 | – | ns |
| 25 | Muxed Address Hold Time | $t_{AHL}$ | 100 | – | 20 | – | ns |
| 26 | Delay Time DS/E to AS/ALE Rise | $t_{ASD}$ | 500 | – | 50 | – | ns |
| 27 | Pulse Width, AS/ALE High | $PW_{ASH}$ | 600 | – | 135 | – | ns |
| 28 | Delay Time, AS/ALE to DS/E Rise | $t_{ASED}$ | 500 | – | 60 | – | ns |
| 30 | Peripheral Output Data Delay Time from DS/E or $\overline{RD}$ | $t_{DDR}$ | 1300 | – | 20 | 240 | ns |
| 31 | Peripheral Data Setup Time | $t_{DSW}$ | 1500 | – | 200 | – | ns |

NOTE: Designations E, ALE, $\overline{RD}$, and $\overline{WR}$ refer to signals from alternative microprocessor signals.

*Refer to IMPORTANT NOTICES appearing on page 20 of this data sheet.

FIGURE 2 — MC146818 BUS TIMING



NOTE: $V_{HIGH}$ = $V_{DD}$ – 2.0 V, $V_{LOW}$ = 0.8 V, for $V_{DD}$ = 5.0 V ± 10%

FIGURE 3 — BUS READ TIMING COMPETITOR MULTIPLEXED BUS

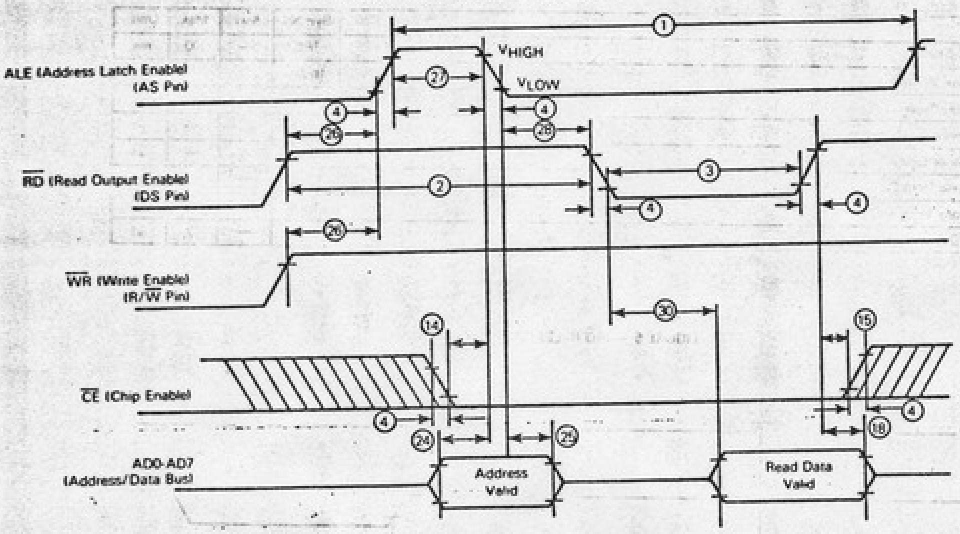
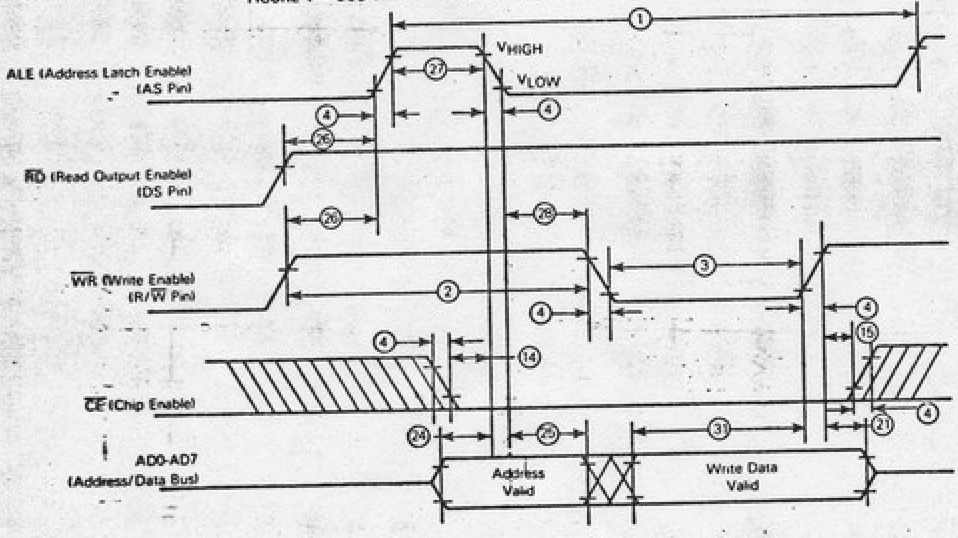


FIGURE 4 — BUS WRITE TIMING COMPETITOR MULTIPLEXED BUS

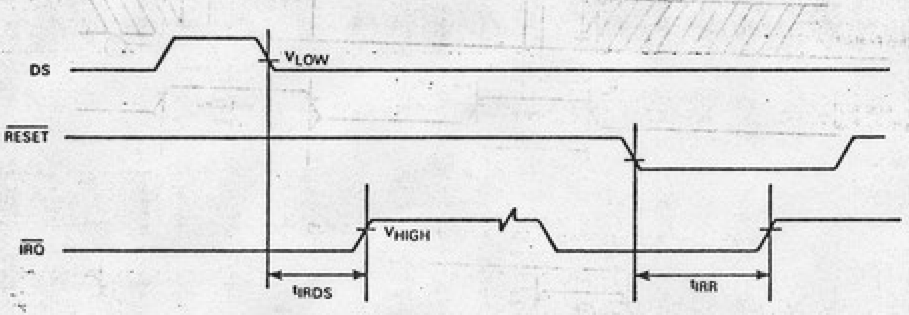


NOTE: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ± 10%

TABLE 1 — SWITCHING CHARACTERISTICS ($V_{DD}$ = 5.0 Vdc ± 10%, $V_{SS}$ = 0 Vdc, $T_A = T_L$ to $T_H$)

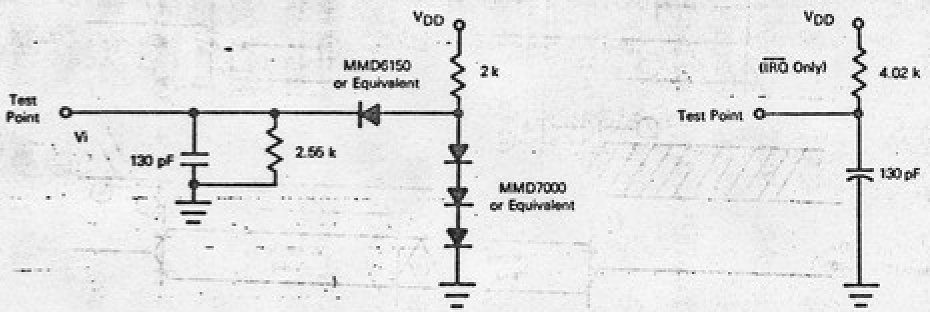
| Description | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| Oscillator Startup | $t_{RC}$ | — | 100 | ms |
| Reset Pulse Width | $t_{RWL}$ | 5 | — | µs |
| Reset Delay Time | $t_{RLH}$ | 5 | — | µs |
| Power Sense Pulse Width | $t_{PWL}$ | 5 | — | µs |
| Power Sense Delay Time | $t_{PLH}$ | 5 | — | µs |
| $\overline{IRQ}$ Release from DS | $t_{IRDS}$ | — | 2 | µs |
| $\overline{IRQ}$ Release from $\overline{RESET}$ | $t_{IRR}$ | — | 2 | µs |
| VRT Bit Delay | $t_{VRTD}$ | — | 2 | µs |

FIGURE 5 — $\overline{IRQ}$ RELEASE DELAY



NOTE: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ± 10%

FIGURE 6 — TTL EQUIVALENT TEST LOAD



All Outputs Except OSC2 (See Figure 10)

FIGURE 7 — POWER-UP

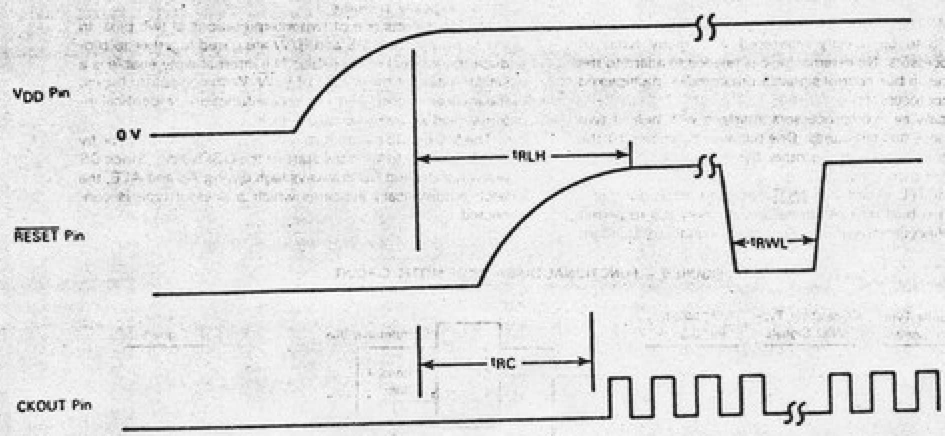
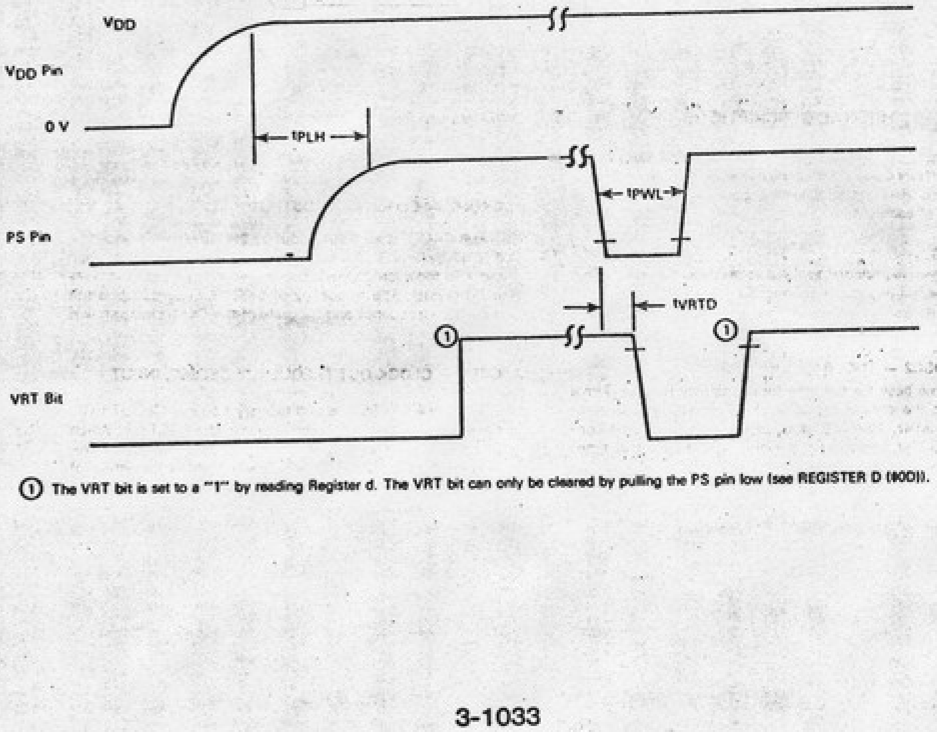


FIGURE 8 — CONDITIONS THAT CLEAR VRT BIT



(1) The VRT bit is set to a "1" by reading Register d. The VRT bit can only be cleared by pulling the PS pin low (see REGISTER D ($0D)).

## MOTEL

The MOTEL circuit is a new concept that permits the MC146818 to be directly interfaced with many types of microprocessors. No external logic is needed to adapt to the differences in bus control signals from common multiplexed bus microprocessors.

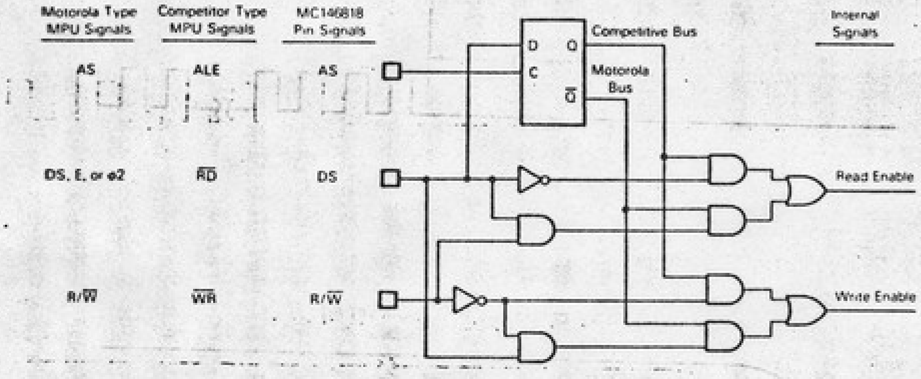
Practically all microprocessors interface with one of two synchronous bus structures. One bus was originated by the Motorola MC6800 and the other by the Intel 8080 and its companion part, the 8228.

The MOTEL circuit (for MOTorola and IntEL bus compatibility) is built into peripheral and memory ICs to permit direct connection to either type of bus. An industry standard bus structure is now available. The MOTEL concept is shown logically in Figure 9.

MOTEL selects one of two interpretations of two pins. In the Motorola case, DS and $R/\overline{W}$ are gated together to produce the internal read enable. The internal write enable is a similar gating of the inverse of $R/\overline{W}$. With competitor buses, the inversion of $\overline{RD}$ and $\overline{WR}$ create functionally identical internal read and write enable signals.

The MC146818 automatically selects the processor type by using AS/ALE to latch the state of the DS/$\overline{RD}$ pin. Since DS is always low and $\overline{RD}$ is always high during AS and ALE, the latch automatically indicates which processor type is connected.

FIGURE 9 — FUNCTIONAL DIAGRAM OF MOTEL CIRCUIT



## SIGNAL DESCRIPTIONS

The block diagram in Figure 1, shows the pin connection with the major internal functions of the MC146818 Real-Time Clock plus RAM. The following paragraphs describe the function of each pin.

### $V_{DD}$, $V_{SS}$

DC power is provided to the part on these two pins, $V_{DD}$ being the more positive voltage. The minimum and maximum voltages are listed in the Electrical Characteristics tables.

### OSC1, OSC2 — TIME BASE, INPUTS

The time base for the time functions may be an external signal or the crystal oscillator. External square waves at 4.194304 MHz, 1.048576 MHz, or 32.768 kHz may be connected to OSC1 as shown in Figure 10. The internal time-base frequency to be used is chosen in Register A.

The on-chip oscillator is designed for a parallel resonant AT cut crystal at 4.194304 MHz or 1.048576 MHz frequencies. The crystal connections are shown in Figure 11 and the crystal characteristics in Figure 12.

### CKOUT — CLOCK OUT, OUTPUT

The CKOUT pin is an output at the time-base frequency divided by 1 or 4. A major use for CKOUT is as the input clock to the microprocessor; thereby saving the cost of a second crystal. The frequency of CKOUT depends upon the time-base frequency and the state of the CKFS pin as shown in Table 2.

### CKFS — CLOCK OUT FREQUENCY SELECT, INPUT

When the CKFS pin is tied to $V_{DD}$ it causes CKOUT to be the same frequency as the time base at the OSC1 pin. When CKFS is tied to $V_{SS}$, CKOUT is the OSC1 time-base frequency divided by four. Table 2 summarizes the effect of CKFS.

FIGURE 10 — EXTERNAL TIME-BASE CONNECTION

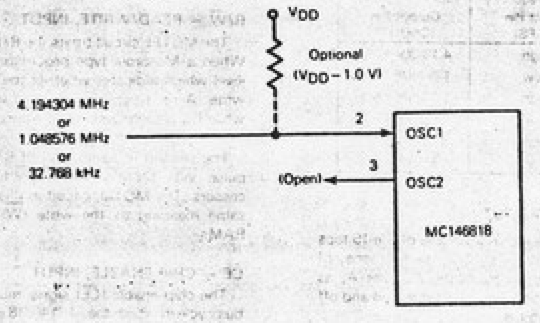
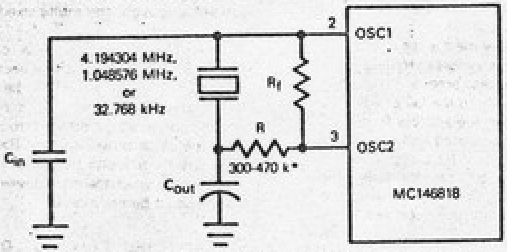


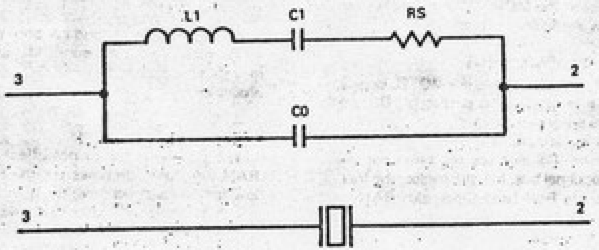
FIGURE 11 — CRYSTAL OSCILLATOR CONNECTION



*32.768 kHz Only — Consult Crystal Manufacturer's Specification

FIGURE 12 — CRYSTAL PARAMETERS

Crystal Equivalent Circuit



| $f_{osc}$ | 4.194304 MHz | 1.048576 MHz | 32.768 kHz |
|---|---|---|---|
| RS (Maximum) | 75 Ω | 700 Ω | 50 k |
| C0 (Maximum) | 7 pF | 5 pF | 1.7 pF |
| C1 | 0.012 pF | 0.008 pF | 0.003 pF |
| Q | 50 k | 35 k | 30 k |
| $C_{in}/C_{out}$ | 15-30 pF | 15-40 pF | 10-22 pF |
| R | – | – | 300-470 k |
| $R_f$ | 10 M | 10 M | 22 M |

TABLE 2 — CLOCK OUTPUT FREQUENCIES

| Time Base (OSC1) Frequency | Clock Frequency Select Pin (CKFS) | Clock Frequency Output Pin (CKOUT) |
|---|---|---|
| 4.194304 MHz | High | 4.194304 MHz |
| 4.194304 MHz | Low | 1.048576 MHz |
| 1.048576 MHz | High | 1.048576 MHz |
| 1.048576 MHz | Low | 262.144 kHz |
| 32.768 kHz | High | 32.768 kHz |
| 32.768 kHz | Low | 8.192 kHz |

### SQW — SQUARE WAVE, OUTPUT

The SQW pin can output a signal from one of the 15 taps provided by the 22 internal-divider stages. The frequency of the SQW may be altered by programming Register A, as shown in Table 5. The SQW signal may be turned on and off using the SQWE bit in Register B.

### AD0-AD7 — MULTIPLEXED BIDIRECTIONAL ADDRESS/DATA BUS

Multiplexed bus processors save pins by presenting the address during the first portion of the bus cycle and using the same pins during the second portion for data. Address-then-data multiplexing does not slow the access time of the MC146818 since the bus reversal from address to data is occurring during the internal RAM access time.

The address must be valid just prior to the fall of AS/ALE at which time the MC146818 latches the address from AD0 to AD5. Valid write data must be presented and held stable during the latter portion of the DS or $\overline{\text{WR}}$ pulses. In a read cycle, the MC146818 outputs eight bits of data during the latter portion of the DS or $\overline{\text{RD}}$ pulses, then ceases driving the bus (returns the output drivers to the high-impedance state) when DS falls in the Motorola case of MOTEL or $\overline{\text{RD}}$ rises in the other case.

### AS — MULTIPLEXED ADDRESS STROBE, INPUT

A positive going multiplexed address strobe pulse serves to demultiplex the bus. The falling edge of AS or ALE causes the address to be latched within the MC146818. The automatic MOTEL circuit in the MC146818 also latches the state of the DS pin with the falling edge of AS or ALE.

### DS — DATA STROBE OR READ, INPUT

The DS pin has two interpretations via the MOTEL circuit. When emanating from a Motorola type processor, DS is a positive pulse during the latter portion of the bus cycle, and is variously called DS (data strobe), E (enable), and $\phi 2$ ($\phi 2$ clock). During read cycles, DS signifies the time that the RTC is to drive the bidirectional bus. In write cycles, the trailing edge of DS causes the Real-Time Clock plus RAM to latch the written data.

The second MOTEL interpretation of DS is that of $\overline{\text{RD}}$, $\overline{\text{MEMR}}$, or $\overline{\text{I/OR}}$ emanating from the competitor type processor. In this case, DS identifies the time period when the real-time clock plus RAM drives the bus with read data. This interpretation of DS is also the same as an output-enable signal on a typical memory.

The MOTEL circuit, within the MC146818, latches the state of the DS pin on the falling edge of AS/ALE. When the Motorola mode of MOTEL is desired DS must be low during AS/ALE, which is the case with the Motorola multiplexed bus processors. To ensure the competitor mode of MOTEL, the DS pin must remain high during the time AS/ALE is high.

### R/$\overline{\text{W}}$ — READ/WRITE, INPUT

The MOTEL circuit treats the R/$\overline{\text{W}}$ pin in one of two ways. When a Motorola type processor is connected, R/$\overline{\text{W}}$ is a level which indicates whether the current cycle is a read or write. A read cycle is indicated with a high level on R/$\overline{\text{W}}$ while DS is high, whereas a write cycle is a low on R/$\overline{\text{W}}$ during DS.

The second interpretation of R/$\overline{\text{W}}$ is as a negative write pulse, $\overline{\text{WR}}$, $\overline{\text{MEMW}}$, and $\overline{\text{I/OW}}$ from competitor type processors. The MOTEL circuit in this mode gives R/$\overline{\text{W}}$ pin the same meaning as the write (W) pulse on many generic RAMs.

### $\overline{\text{CE}}$ — CHIP ENABLE, INPUT

The chip-enable ($\overline{\text{CE}}$) signal must be asserted (low) for a bus cycle in which the MC146818 is to be accessed. $\overline{\text{CE}}$ is not latched and must be stable during DS and AS (Motorola case of MOTEL) and during $\overline{\text{RD}}$ and $\overline{\text{WR}}$ (in the other MOTEL case). Bus cycles which take place without asserting $\overline{\text{CE}}$ cause no actions to take place within the MC146818. When $\overline{\text{CE}}$ is high, the multiplexed bus output is in a high-impedance state.

When $\overline{\text{CE}}$ is high, all address, data, DS, and R/$\overline{\text{W}}$ inputs from the processor are disconnected within the MC146818. This permits the MC146818 to be isolated from a powered-down processor. When $\overline{\text{CE}}$ is held high, an unpowered device cannot receive power through the input pins from the real-time clock power source. Battery power consumption can thus be reduced by using a pullup resistor or active clamp on $\overline{\text{CE}}$ when the main power is off. When $\overline{\text{CE}}$ is not used, it should be grounded.

### $\overline{\text{IRQ}}$ — INTERRUPT REQUEST, OUTPUT

The $\overline{\text{IRQ}}$ pin is an active low output of the MC146818 that may be used as an interrupt input to a processor. The $\overline{\text{IRQ}}$ output remains low as long as the status bit causing the interrupt is present and the corresponding interrupt-enable bit is set. To clear the $\overline{\text{IRQ}}$ pin, the processor program normally reads Register C. The $\overline{\text{RESET}}$ pin also clears pending interrupts.

When no interrupt conditions are present, the $\overline{\text{IRQ}}$ level is in the high-impedance state. Multiple interrupting devices may thus be connected to an $\overline{\text{IRQ}}$ bus with one pullup at the processor.

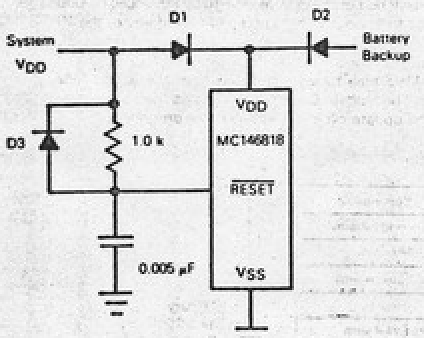
### $\overline{\text{RESET}}$ — RESET, INPUT

The $\overline{\text{RESET}}$ pin does not affect the clock, calendar, or RAM functions. On powerup, the $\overline{\text{RESET}}$ pin must be held low for the specified time, $t_{RLH}$, in order to allow the power supply to stabilize. Figure 13 shows a typical representation of the $\overline{\text{RESET}}$ pin circuit.

When $\overline{\text{RESET}}$ is low the following occurs:

a) Periodic Interrupt Enable (PIE) bit is cleared to zero,
b) Alarm Interrupt Enable (AIE) bit is cleared to zero,
c) Update ended Interrupt Enable (UIE) bit is cleared to zero,
d) Update ended Interrupt Flag (UF) bit is cleared to zero,
e) Interrupt Request status Flag (IRQF) bit is cleared to zero,
f) Periodic Interrupt Flag (PF) bit is cleared to zero,
g) The part is not accessible.

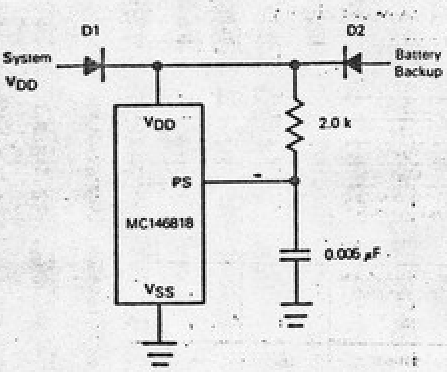
FIGURE 13 — TYPICAL POWERUP DELAY CIRCUIT FOR $\overline{\text{RESET}}$



D1 = MBD701 (Schottky) or Equivalent
D2 = D3 = 1N4148 or Equivalent

Note: If the RTC is isolated from the MPU or MCU power by a diode drop, care must be taken to meet $V_{in}$ requirements.

FIGURE 14 — TYPICAL POWERUP DELAY CIRCUIT FOR POWER SENSE



D1 = MBD701 (Schottky) or Equivalent
D2 = 1N4148 or Equivalent

g) Alarm Interrupt Flag (AF) bit is cleared to zero,
h) $\overline{\text{IRQ}}$ pin is in high-impedance state, and
i) Square Wave output Enable (SQWE) bit is cleared to zero.

PS — POWER SENSE, INPUT

The power-sense pin is used in the control of the valid RAM and time (VRT) bit in Register D. When the PS pin is low the VRT bit is cleared to zero.

When using the VRT feature during powerup, the PS pin must be externally held low for the specified $t_{PLH}$ time. As power is applied, the VRT bit remains low indicating that the contents of the RAM, time registers, and calendar are not guaranteed. PS must go high after powerup to allow the VRT bit to be set by a read of register D.

## POWER-DOWN CONSIDERATIONS

In most systems, the MC146818 must continue to keep time when system power is removed. In such systems, a conversion from system power to an alternate power supply, usually a battery, must be made. During the transition from system to battery power, the designer of a battery backed-up RTC system must protect data integrity, minimize power consumption, and ensure hardware reliability.

The chip enable ($\overline{\text{CE}}$) pin controls all bus inputs (R/$\overline{\text{W}}$, DS, AS, AD0-AD7). $\overline{\text{CE}}$, when negated, disallows any unintended modification of the RTC data by the bus. $\overline{\text{CE}}$ also reduces power consumption by reducing the number of transitions seen internally.

Power consumption may be further reduced by removing resistive and capacitive loads from the clock out (CKOUT) pin and the squarewave (SQW) pin.

During and after the power source conversion, the $V_{IN}$ maximum specification must never be exceeded. Failure to meet the $V_{IN}$ maximum specification can cause a virtual SCR to appear which may result in excessive current drain and destruction of the part.

## ADDRESS MAP

Figure 15 shows the address map of the MC146818. The memory consists of 50 general purpose RAM bytes, 10 RAM bytes which normally contain the time, calendar, and alarm data, and four control and status bytes. All 64 bytes are directly readable and writable by the processor program except for the following: 1) Registers C and D are read only, 2) bit 7 of Register A is read only, and 3) the high-order bit of the seconds byte is read only. The contents of four control and status registers (A, B, C, and D) are described in **REGISTERS.**

### TIME, CALENDAR, AND ALARM LOCATIONS

The processor program obtains time and calendar information by reading the appropriate locations. The program may initialize the time, calendar, and alarm by writing to these RAM locations. The contents of the 10 time, calendar, and alarm bytes may be either binary or binary-coded decimal (BCD).

Before initializing the internal registers, the SET bit in Register B should be set to a "1" to prevent time/calendar updates from occurring. The program initializes the 10 locations in the selected format (binary or BCD), then indicates the format in the data mode (DM) bit of Register B. All 10 time, calendar, and alarm bytes must use the same data mode, either binary or BCD. The SET bit may now be cleared to allow updates. Once initialized the real-time clock makes all updates in the selected data mode. The data mode cannot be changed without reinitializing the 10 data bytes.

Table 3 shows the binary and BCD formats of the 10 time, calendar, and alarm locations. The 24/12 bit in Register B establishes whether the hour locations represent 1-to-12 or 0-to-23. The 24/12 bit cannot be changed without reinitializing the hour locations. When the 12-hour format is selected the high-order bit of the hours byte represents PM when it is a "1".

The time, calendar, and alarm bytes are not always accessable by the processor program. Once-per-second the 10 bytes are switched to the update logic to be advanced by one second and to check for an alarm condition. If any of the 10 bytes are read at this time, the data outputs are undefined. The update lockout time is 248 µs at the 4.194304 MHz and 1.048567 MHz time bases and 1948 µs for the 32.768 kHz time base. The Update Cycle section shows how to accommodate the update cycle in the processor program.

FIGURE 15 — ADDRESS MAP

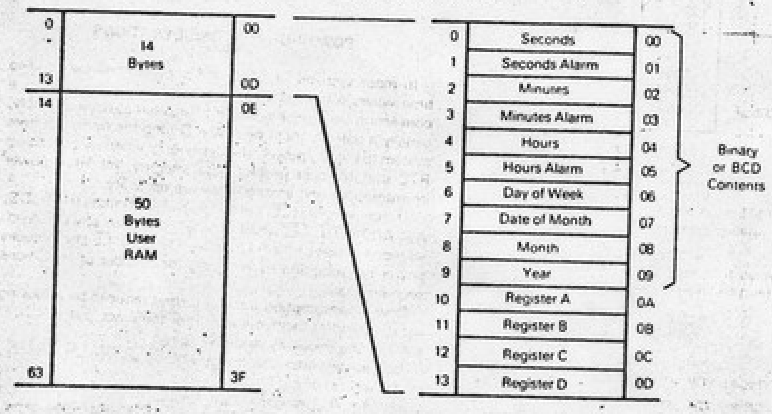


TABLE 3 — TIME, CALENDAR, AND ALARM DATA MODES

| Address Location | Function | Decimal Range | Range |  | Example* |  |
|---|---|---|---|---|---|---|
|  |  |  | Binary Data Mode | BCD Data Mode | Binary Data Mode | BCD Data Mode |
| 0 | Seconds | 0-59 | $00-$3B | $00-$59 | 15 | 21 |
| 1 | Seconds Alarm | 0-59 | $00-$3B | $00-$59 | 15 | 21 |
| 2 | Minutes | 0-59 | $00-$3B | $00-$59 | 3A | 58 |
| 3 | Minutes Alarm | 0-59 | $00-$3B | $00-$59 | 3A | 58 |
| 4 | Hours (12 Hour Mode) | 1-12 | $01-$0C (AM) and $81-$8C (PM) | $01-$12 (AM) and $81-$92 (PM) | 05 | 05 |
|  | Hours (24 Hour Mode) | 0-23 | $00-$17 | $00-$23 | 05 | 05 |
| 5 | Hours Alarm (12 Hour Mode) | 1-12 | $01-$0C (AM) and $81-$8C (PM) | $01-$12 (AM) and $81-$92 (PM) | 05 | 05 |
|  | Hours Alarm (24 Hour Mode) | 0-23 | $00-$17 | $00-23 | 05 | 05 |
| 6 | Day of the Week Sunday = 1 | 1-7 | $01-$07 | $01-$07 | 05 | 05 |
| 7 | Date of the Month | 1-31 | $01-$1F | $01-$31 | 0F | 15 |
| 8 | Month | 1-12 | $01-$0C | $01-$12 | 02 | 02 |
| 9 | Year | 0-99 | $00-$63 | $00-$99 | 4F | 79 |

*Example: 5:58:21 Thursday 15 February 1979 (time is AM)

The three alarm bytes may be used in two ways. First, when the program inserts an alarm time in the appropriate hours, minutes, and seconds alarm locations, the alarm interrupt is initiated at the specified time each day if the alarm enable bit is high. The second usage is to insert a "don't care" state in one or more of three alarm bytes. The "don't care" code is any hexadecimal byte from C0 to FF. That is, the two most-significant bits of each byte, when set to "1", create a "don't care" situation. An alarm interrupt each hour is created with a "don't care" code in the hours alarm location. Similarly, an alarm is generated every minute with "don't care" codes in the hours and minutes alarm bytes. The "don't care" codes in all three alarm bytes create an interrupt every second.

## STATIC CMOS RAM

The 50 general purpose RAM bytes are not dedicated within the MC146818. They can be used by the processor program, and are fully available during the update cycle.

When time and calendar information must use battery back-up, very frequently there is other non-volatile data that must be retained when main power is removed. The 50 user RAM bytes serve the need for low-power CMOS battery-backed storage, and extend the RAM available to the program.

When further CMOS RAM is needed, additional MC146818s may be included in the system. The time/calendar functions may be disabled by holding the DV0-DV2 dividers, in Register A, in the reset state by setting the SET bit in Register B or by removing the oscillator. Holding the dividers in reset prevents interrupts or SQW output from operating while setting the SET bit allows these functions to occur. With the dividers clear, the available user RAM is extended to 59 bytes. The high-order bit of the seconds byte, bit 7 of Register A, and all bits of Registers C and D cannot effectively be used as general purpose RAM.

## INTERRUPTS

The RTC plus RAM includes three separate fully automatic sources of interrupts to the processor. The alarm interrupt may be programmed to occur at rates from once-per-second to one-a-day. The periodic interrupt may be selected for rates from half-a-second to 30.517 μs. The update-ended interrupt may be used to indicate to the program that an update cycle is completed. Each of these independent interrupt conditions are described in greater detail in other sections.

The processor program selects which interrupts, if any, it wishes to receive. Three bits in Register B enable the three interrupts. Writing a "1" to a interrupt-enable bit permits that interrupt to be initiated when the event occurs. A "0" in the interrupt-enable bit prohibits the IRQ pin from being asserted due to the interrupt cause.

If an interrupt flag is already set when the interrupt becomes enabled, the $\overline{\text{IRQ}}$ pin is immediately activated, though the interrupt initiating the event may have occurred much earlier. Thus, there are cases where the program should clear such earlier initiated interrupts before first enabling new interrupts.

When an interrupt event occurs a flag bit is set to a "1" in Register C. Each of the three interrupt sources have separate flag bits in Register C, which are set independent of the state of the corresponding enable bits in Register B. The flag bit may be used with or without enabling the corresponding enable bits.

In the software scanned case, the program does not enable the interrupt. The "interrupt" flag bit becomes a status bit, which the software interrogates, when it wishes. When the software detects that the flag is set, it is an indication to software that the "interrupt" event occurred since the bit was last read.

However, there is one precaution. The flag bits in Register C are cleared (record of the interrupt event is erased) when Register C is read. Double latching is included with Register C so the bits which are set are stable throughout the read cycle. All bits which are high when read by the program are cleared, and new interrupts (on any bits) are held until after the read cycle. One, two, or three flag bits may be found to be set when Register C is read. The program should inspect all utilized flag bits every time Register C is read to insure that no interrupts are lost.

The second flag bit usage method is with fully enabled interrupts. When an interrupt-flag bit is set and the corresponding interrupt-enable bit is also set, the $\overline{\text{IRQ}}$ pin is asserted low. $\overline{\text{IRQ}}$ is asserted as long as at least one of the three interrupt sources has its flag and enable bits both set. The IRQF bit in Register C is a "1" whenever the $\overline{\text{IRQ}}$ pin is being driven low.

The processor program can determine that the RTC initiated the interrupt by reading Register C. A "1" in bit 7 (IRQF bit) indicates that one or more interrupts have been initiated by the part. The act of reading Register C clears all the then-active flag bits, plus the IRQF bit. When the program finds IRQF set, it should look at each of the individual flag bits in the same byte which have the corresponding interrupt-mask bits set and service each interrupt which is set. Again, more than one interrupt-flag bit may be set.

## DIVIDER STAGES

The MC146818 has 22 binary-divider stages following the time base as shown in Figure 1. The output of the dividers is a 1 Hz signal to the update-cycle logic. The dividers are controller by three divider bus (DV2, DV1, and DV0) in Register A.

## DIVIDER CONTROL

The divider-control bits have three uses, as shown in Table 4. Three usable operating time bases may be selected (4.194304 MHz, 1.048576 MHz, or 32.768 kHz). The divider chain may be held reset, which allows precision setting of the time. When the divider is changed from reset to an operating time base, the first update cycle is one-half second later. The divider-control bits are also used to facilitate testing the MC146818.

TABLE 4 — DIVIDER CONFIGURATIONS

| Time-Base Frequency | Divider Bits Register A |  |  | Operation Mode | Divider Reset | Bypass First N-Divider Bits |
|---|---|---|---|---|---|---|
|  | DV2 | DV1 | DV0 |  |  |  |
| 4 194304 MHz | 0 | 0 | 0 | Yes | – | N = 0 |
| 1.048576 MHz | 0 | 0 | 1 | Yes | – | N = 2 |
| 32.768 kHz | 0 | 1 | 0 | Yes | – | N = 7 |
| Any | 1 | 1 | 0 | No | Yes | – |
| Any | 1 | 1 | 1 | No | Yes | – |

Note: Other combinations of divider bits are used for test purposes only.

## SQUARE-WAVE OUTPUT SELECTION

Fifteen of the 22 divider taps are made available to a 1-of-15 selector as shown in Figure 1. The first purpose of selecting a divider tap is to generate a square-wave output signal at the SQW pin. The RS0-RS3 bits in Register A establish the square-wave frequency as listed in Table 5. The SQW frequency selection shares the 1-of-15 selector with periodic interrupts.

Once the frequency is selected, the output of the SQW pin may be turned on and off under program control with the square-wave enable (SQWE) bit in Register B. Altering the divider, square-wave output selection bits, or the SQWE output-enable bit may generate an asymmetrical waveform at the time of execution. The square-wave output pin has a number of potential uses. For example, it can serve as a frequency standard for external use, a frequency synthesizer, or could be used to generate one or more audio tones under program control.

## PERIODIC INTERRUPT SELECTION

The periodic interrupt allows the $\overline{IRQ}$ pin to be triggered from once every 500 ms to once every 30.517 μs. The periodic interrupt is separate from the alarm interrupt which may be output from once-per-second to once-per-day.

Table 5 shows that the periodic interrupt rate is selected with the same Register A bits which select the square-wave frequency. Changing one also changes the other. But each function may be separately enabled so that a program could switch between the two features or use both. The SQW pin is enabled by the SQWE bit in Register B. Similarly the periodic interrupt is enabled by the PIE bit in Register B.

Periodic interrupt is usable by practically all real-time systems. It can be used to scan for all forms of inputs from contact closures to serial receive bits or bytes. It can be used in multiplexing displays or with software counters to measure inputs, create output intervals, or await the next needed software function.

TABLE 5 — PERIODIC INTERRUPT RATE AND SQUARE WAVE OUTPUT FREQUENCY

| Select Bits Register A |  |  |  | 4,194304 or 1.048576 MHz Time Base |  | 32.768 kHz Time Base |  |
|---|---|---|---|---|---|---|---|
| RS3 | RS2 | RS1 | RS0 | Periodic Interrupt Rate $t_{PI}$ | SQW Output Frequency | Periodic Interrupt Rate $t_{PI}$ | SQW Output Frequency |
| 0 | 0 | 0 | 0 | None | None | None | None |
| 0 | 0 | 0 | 1 | 30.517 μs | 32.768 kHz | 3.90625 ms | 256 Hz |
| 0 | 0 | 1 | 0 | 61.035 μs | 16.384 kHz | 7.8125 ms | 128 Hz |
| 0 | 0 | 1 | 1 | 122.070 μs | 8.192 kHz | 122.070 μs | 8.192 kHz |
| 0 | 1 | 0 | 0 | 244.141 μs | 4.096 kHz | 244.141 μs | 4.096 kHz |
| 0 | 1 | 0 | 1 | 488.281 μs | 2.048 kHz | 488.281 μs | 2.048 kHz |
| 0 | 1 | 1 | 0 | 976.562 μs | 1.024 kHz | 976.562 μs | 1.024 kHz |
| 0 | 1 | 1 | 1 | 1.953125 ms | 512 Hz | 1.953125 ms | 512 Hz |
| 1 | 0 | 0 | 0 | 3.90625 ms | 256 Hz | 3.90625 ms | 256 Hz |
| 1 | 0 | 0 | 1 | 7.8125 ms | 128 Hz | 7.8125 ms | 128 Hz |
| 1 | 0 | 1 | 0 | 15.625 ms | 64 Hz | 15.625 ms | 64 Hz |
| 1 | 0 | 1 | 1 | 31.25 ms | 32 Hz | 31.25 ms | 32 Hz |
| 1 | 1 | 0 | 0 | 62.5 ms | 16 Hz | 62.5 ms | 16 Hz |
| 1 | 1 | 0 | 1 | 125 ms | 8 Hz | 125 ms | 8 Hz |
| 1 | 1 | 1 | 0 | 250 ms | 4 Hz | 250 ms | 4 Hz |
| 1 | 1 | 1 | 1 | 500 ms | 2 Hz | 500 ms | 2 Hz |

## UPDATE CYCLE

The MC146818 executes an update cycle once-per-second, assuming one of the proper time bases is in place, the DV0-DV2 divider is not clear, and the SET bit in Register B is clear. The SET bit in the "1" state permits the program to initialize the time and calendar bytes by stopping an existing update and preventing a new one from occurring.

The primary function of the update cycle is to increment the seconds byte, check for overflow, increment the minutes byte when appropriate and so forth through to the year of the century byte. The update cycle also compares each alarm byte with the corresponding time byte and issues an alarm if a match or if a "don't care" code (11XXXXXX) is present in all three positions.

With a 4.194304 MHz or 1.048576 MHz time base the update cycle takes 248 µs while a 32.768 kHz time base update cycle takes 1984 µs. During the update cycle, the time, calendar, and alarm bytes are not accessable by the processor program. The MC146818 protects the program from reading transitional data. This protection is provided by switching the time, calendar, and alarm portion of the RAM off the microprocessor bus during the entire update cycle. If the processor reads these RAM locations before the update is complete the output will be undefined. The update in progress (UIP) status bit is set during the interval.

A program which randomly accesses the time and date information finds data unavailable statistically once every 4032 attempts. Three methods of accommodating nonavailability during update are usable by the program. In discussing the three methods it is assumed that at random points user programs are able to call a subroutine to obtain the time of day.

The first method of avoiding the update cycle uses the update-ended interrupt. If enabled, an interrupt occurs after every update cycle which indicates that over 999 ms are available to read valid time and date information. During this time a display could be updated or the information could be transfered to continuously available RAM. Before leaving the interrupt sevice routine, the IRQF bit in Register C should be cleared.

The second method uses the update-in-progress bit (UIP) in Register A to determine if the update cycle is in progress or not. The UIP bit will pulse once-per-second. Statistically, the UIP bit will indicate that time and date information is unavailable once every 2032 attempts. After the UIP bit goes high, the update cycle begins 244 µs later. Therefore, if a low is read on the UIP bit, the user has at least 244 µs before the time/calendar data will be changed. If a "1" is read in the UIP bit, the time/calendar data may not be valid. The user should avoid interrupt service routines that would cause the time needed to read valid time/calendar data to exceed 244 µs.

The third method uses a periodic interrupt to determine if an update cycle is in progress. The UIP bit in Register A is set high between the setting of the PF bit in Register C (see Figure 16). Periodic interrupts that occur at a rate of greater than $t_{BUC} + t_{UC}$ allow valid time and date information to be read at each occurrence of the periodic interrupt. The reads should be completed within $(T_{PI} \div 2) + t_{BUC}$ to ensure that data is not read during the update cycle.

To properly setup the internal counters for daylight savings time operation, the user must set the time at least two seconds before the rollover will occur. Likewise, the time must be set at least two seconds before the end of the 29th or 30th day of the month.

## REGISTERS

The MC146818 has four registers which are accessible to the processor program. The four registers are also fully accessible during the update cycle.

### REGISTER A ($0A)

MSB ... LSB

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| UIP | DV2 | DV1 | DV0 | RS3 | RS2 | RS1 | RS0 |

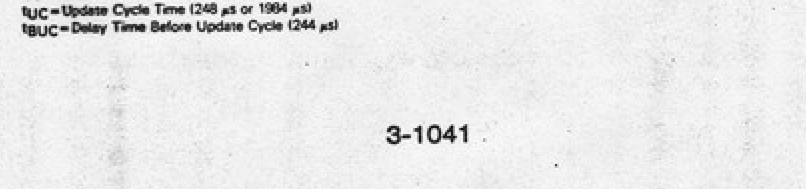
Read/Write Register except UIP

UIP — The update in progress (UIP) bit is a status flag that may be monitored by the program. When UIP is a "1" the update cycle is in progress or will soon begin. When UIP is a "0" the update cycle is not in progress and will not be for at least 244 µs (for all time bases). This is detailed in Table 6. The time, calendar, and alarm information in RAM is fully available to the program when the UIP bit is zero — it is not in transition. The UIP bit is a read-only bit, and is not affected by Reset. Writing the SET bit in Register B to a "1" inhibit any update cycle and then clear the UIP status bit.

TABLE 6 — UPDATE CYCLE TIMES

| UIP Bit | Time Base (OSC1) | Update Cycle Time ($t_{UC}$) | Minimum Time Before Update Cycle ($t_{BUC}$) |
|---|---|---|---|
| 1 | 4.194304 MHz | 248 µs | – |
| 1 | 1.048576 MHz | 248 µs | – |
| 1 | 32.768 kHz | 1984 µs | – |
| 0 | 4.194304 MHz | – | 244 µs |
| 0 | 1.048576 MHz | – | 244 µs |
| 0 | 32.768 kHz | – | 244 µs |

FIGURE 16 — UPDATE-ENDED AND PERIODIC INTERRUPT RELATIONSHIPS



$t_{PI}$ = Periodic Interrupt Time Interval (500 ms, 250 ms, 125 ms, 62.5 ms, etc. per Table 5)
$t_{UC}$ = Update Cycle Time (248 µs or 1984 µs)
$t_{BUC}$ = Delay Time Before Update Cycle (244 µs)

**DV2, DV1, DV0** — Three bits are used to permit the program to select various conditions of the 22-stage divider chain. The divider selection bits identify which of the three time-base frequencies is in use. Table 4 shows that time bases of 4.194304 MHz, 1.048576 MHz, and 32.768 kHz may be used. The divider selection bits are also used to reset the divider chain. When the time/calendar is first initialized, the program may start the divider at the precise time stored in the RAM. When the divider reset is removed the first update cycle begins one-half second later. These three read/write bits are not affected by $\overline{\text{RESET}}$.

**RS3, RS2, RS1, RS0** — The four rate selection bits select one of 15 taps on the 22-stage divider, or disable the divider output. The tap selected may be used to generate an output square wave (SQW pin) and/or a periodic interrupt. The program may do one of the following: 1) enable the interrupt with the PIE bit, 2) enable the SQW output pin with the SQWE bit, 3) enable both at the same time at the same rate, or 4) enable neither. Table 5 lists the periodic interrupt rates and the square-wave frequencies that may be chosen with the RS bits. These four bits are read/write bits which are not affected by $\overline{\text{RESET}}$

**REGISTER B ($0B)**

MSB … LSB — Read/Write Register

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| SET | PIE | AIE | UIE | SQWE | DM | 24/12 | DSE |

**SET** — When the SET bit is a "0", the update cycle functions normally by advancing the counts once-per-second. When the SET bit is written to a "1", any update cycle in progress is aborted and the program may initialize the time and calendar bytes without an update occurring in the midst of initializing. SET is a read/write bit which is not modified by $\overline{\text{RESET}}$ or internal functions of the MC146818.

**PIE** — The periodic interrupt enable (PIE) bit is a read/write bit which allows the periodic-interrupt flag (PF) bit in Register C to cause the $\overline{\text{IRQ}}$ pin to be driven low. A program writes a "1" to the PIE bit in order to receive periodic interrupts at the rate specified by the RS3, RS2, RS1, and RS0 bits in Register A. A zero in PIE blocks $\overline{\text{IRQ}}$ from being initiated by a periodic interrupt, but the periodic flag (PF) bit is still set at the periodic rate. PIE is not modified by any internal MC146818 functions, but is cleared to "0" by a $\overline{\text{RESET}}$.

**AIE** — The alarm interrupt enable (AIE) bit is a read/write bit which when set to a "1" permits the alarm flag (AF) bit in Register C to assert $\overline{\text{IRQ}}$. An alarm interrupt occurs for each second that the three time bytes equal the three alarm bytes (including a "don't care" alarm code of binary 11XXXXXX). When the AIE bit is a "0", the AF bit does not initiate an $\overline{\text{IRQ}}$ signal. The $\overline{\text{RESET}}$ pin clears AIE to "0". The internal functions do not affect the AIE bit.

**UIE** — The UIE (update-ended interrupt enable) bit is a read/write bit which enables the update-end flag (UF) bit in Register C to assert $\overline{\text{IRQ}}$. The $\overline{\text{RESET}}$ pin going low or the SET bit going high clears the UIE bit.

**SQWE** — When the square-wave enable (SQWE) bit is set to a "1" by the program, a square-wave signal at the frequency specified in the rate selection bits (RS3 to RS0) appears on the SQW pin. When the SQWE bit is set to a zero the SQW pin is held low. The state of SQWE is cleared by the $\overline{\text{RESET}}$ pin. SQWE is a read/write bit.

**DM** — The data mode (DM) bit indicates whether time and calendar updates are to use binary or BCD formats. The DM bit is written by the processor program and may be read by the program, but is not modified by any internal functions or $\overline{\text{RESET}}$. A "1" in DM signifies binary data, while a "0" in DM specifies binary-coded-decimal (BCD) data.

**24/12** — The 24/12 control bit establishes the format of the hours bytes as either the 24-hour mode (a "1") or the 12-hour mode (a "0"). This is a read/write bit, which is affected only by software.

**DSE** — The daylight savings enable (DSE) bit is a read/write bit which allows the program to enable two special updates (when DSE is a "1"). On the last Sunday in April the time increments from 1:59:59 AM to 3:00:00 AM. On the last Sunday in October when the time first reaches 1:59:59 AM it changes to 1:00:00 AM. These special updates do not occur when the DSE bit is a "0". DSE is not changed by any internal operations or reset.

**REGISTER C ($0C)**

MSB … LSB — Read-Only Register

| b7 | b6 | b5 | b4 | b3 | b | b1 | b0 |
|---|---|---|---|---|---|---|---|
| IRQF | PF | AF | UF | 0 | 0 | 0 | 0 |

**IRQF** — The interrupt request flag (IRQF) is set to a "1" when one or more of the following are true:

PF = PIE = "1"

AF = AIE = "1"

UF = UIE = "1"

i.e., IRQF = PF•PIE + AF•AIE + UF•UIE

Any time the IRQF bit is a "1", the $\overline{\text{IRQ}}$ pin is driven low. All flag bits are cleared after Register C is read by the program or when the $\overline{\text{RESET}}$ pin is low.

**PF** — The periodic interrupt flag (PF) is a read-only bit which is set to a "1" when a particular edge is detected on the selected tap of the divider chain. The RS3 to RS0 bits establish the periodic rate. PF is set to a "1" independent of the state of the PIE bit. PF being a "1" initiates an $\overline{\text{IRQ}}$ signal and sets the IRQF bit when PIE is also a "1". The PF bit is cleared by a $\overline{\text{RESET}}$ or a software read of Register C.

**AF** — A "1" in the AF (alarm interrupt flag) bit indicates that the current time has matched the alarm time. A "1" in the AF causes the $\overline{\text{IRQ}}$ pin to go low, and a "1" to appear in the IRQF bit, when the AIE bit also is a "1." A $\overline{\text{RESET}}$ or a read of Register C clears AF.

**UF** — The update-ended interrupt flag (UF) bit is set after each update cycle. When the UIE bit is a "1", the "1" in UF causes the IRQF bit to be a "1", asserting $\overline{\text{IRQ}}$. UF is cleared by a Register C read or a $\overline{\text{RESET}}$.

**b3 TO b0** — The unused bits of Status Register 1 are read as "0's". They can not be written.

**REGISTER D ($0D)**

MSB LSB

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| VRT | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Read Only Register

**VRT** — The valid RAM and time (VRT) bit indicates the condition of the contents of the RAM, provided the power sense (PS) pin is satisfactorily connected. A "0" appears in the VRT bit when the power-sense pin is low. The processor program can set the VRT bit when the time and calendar are initialized to indicate that the RAM and time are valid. The VRT is a read only bit which is not modified by the $\overline{\text{RESET}}$ pin. The VRT bit can only be set by reading Register D.

**b6 TO b0** — The remaining bits of Register D are unused. They cannot be written, but are always read as "0's."

## TYPICAL INTERFACING

The MC146818 is best suited for use with microprocessors which generate an address-then-data multiplexed bus. Figures 17 and 18 show typical interfaces to bus-compatible processors. These interfaces assume that the address decoding can be done quickly. However, if standard metal-gate CMOS gates are used the $\overline{\text{CE}}$ setup time may be violated. Figure 19 illustrates an alternative method of chip selection which will accommodate such slower decoding.

The MC146818 can be interfaced to single-chip microcomputers (MCU) by using eleven port lines as shown in Figure 20. Non-multiplexed bus microprocessors can be interfaced with additional support.

There is one method of using the multiplexed bus MC146818 with non-multiplexed bus processors. The interface uses available bus control signals to multiplex the address and data bus together.

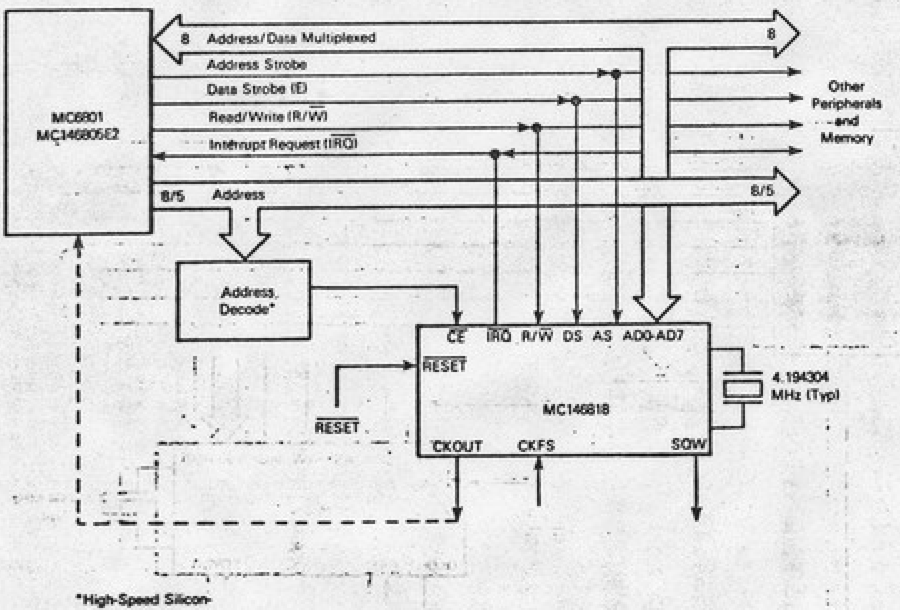
An example using either the Motorola MC6800, MC6802, MC6808, or MC6809 microprocessor is shown in Figure 21.

Figure 22 illustrates the subroutines which may be used for data transfers in a non-multiplexed system. The subroutines should be entered with the registers containing the following data:

Accumulator A: The address of the RTC to be accessed.
Accumulator B: Write: The data to be written.
Read: The data read from the RTC.

The RTC is mapped to two consecutive memory locations — RTC and RTC + 1 as shown in Figure 21.

FIGURE 17 — MC146818 INTERFACED WITH MOTOROLA COMPATIBLE MULTIPLEXED BUS MICROPROCESSORS



*High-Speed Silicon-Gate CMOS or TTL Address Decoding

FIGURE 18 — MC146818 INTERFACED WITH COMPETITOR COMPATIBLE MULTIPLEXED BUS MICROPROCESSORS

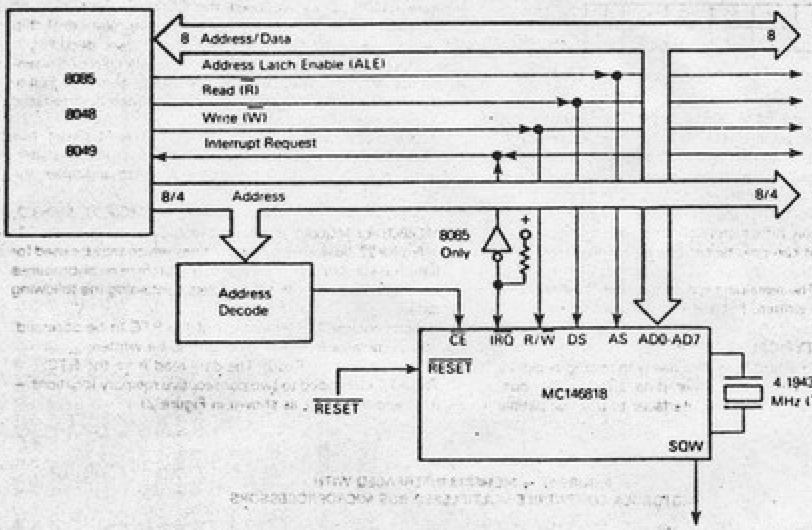
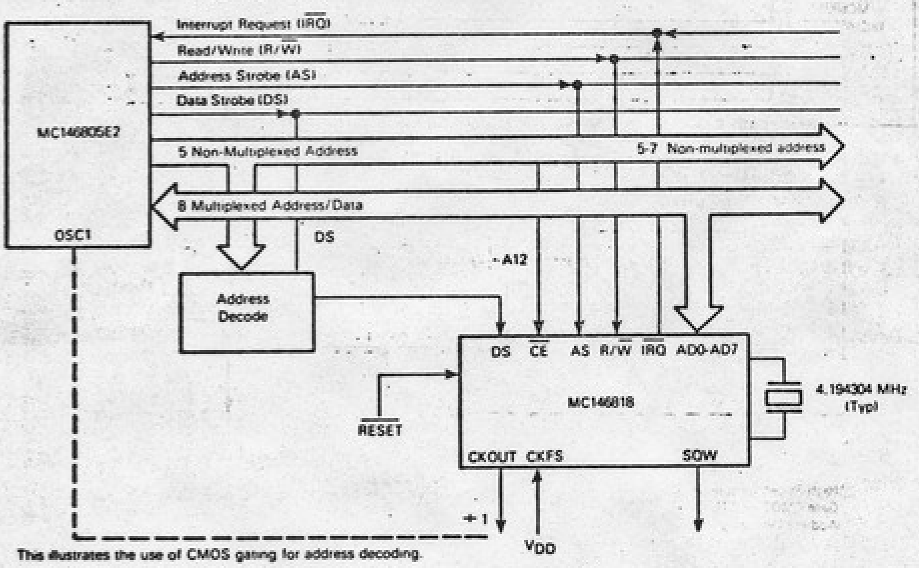


FIGURE 19 — MC146818 INTERFACE WITH MC146805E2 CMOS MULTIPLEXED MICROPROCESSOR WITH SLOW ADDRESSING DECODING



This illustrates the use of CMOS gating for address decoding.

FIGURE 20 — MC146818 INTERFACED WITH THE PORTS OF A TYPICAL SINGLE CHIP MICROCOMPUTER

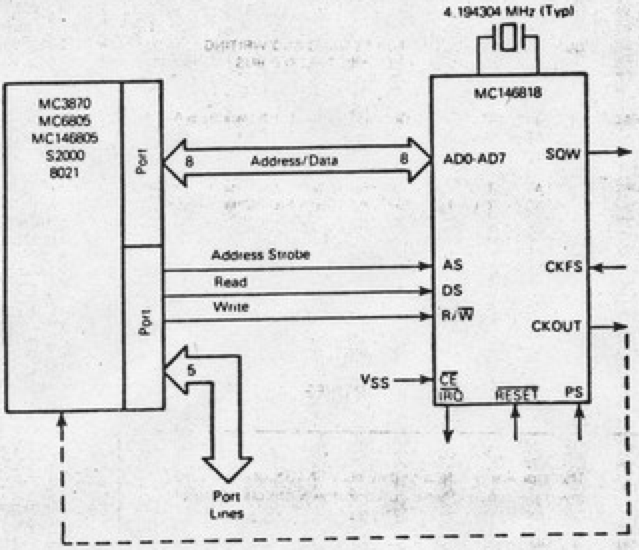


FIGURE 21 — MC146818 INTERFACED WITH MOTOROLA PROCESSORS

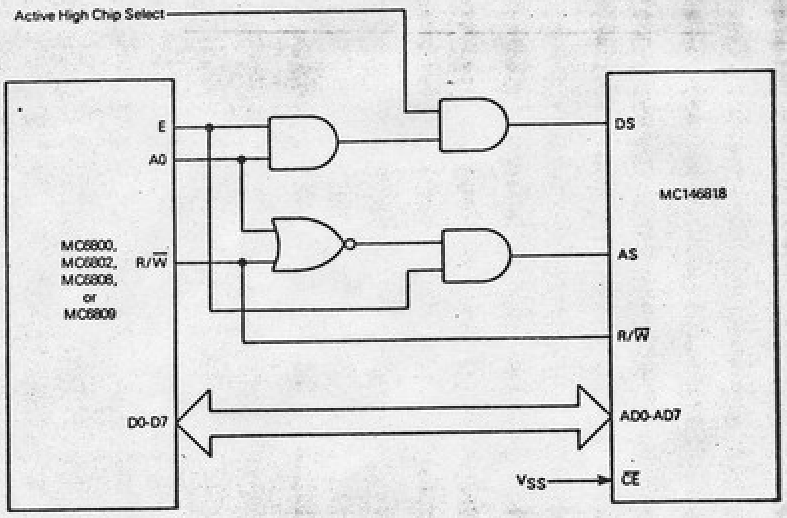


FIGURE 22 — SUBROUTINE FOR READING AND WRITING THE MC146818 WITH A NON-MULTIPLEXED BUS

```
READ    STA     RTC          Generate AS and Latch Data from ACCA
        LDAB    RTC+1        Generate DS and Get Data
        RTS

WRITE   STA     RTC          Generate AS and Latch Data from ACCA
        STAB    RTC+1        Generate DS and Store Data
        RTS
```

## IMPORTANT NOTICES

These devices made with date code 3N4GXXXX have the following exceptions when used in the Motorola mode of MOTEL.

1. $V_{DD}$ = 3 to 5.25 V for operation
2. DS $V_{IL}$ = 0.6 V Max.

The falling edge of chip select should occur during the active high pulse of address strobe, only on those units with date code CC6XXXX.