Rockwell

R65C00/21 DUAL CMOS MICROCOMPUTER AND R65C29 DUAL CMOS MICROPROCESSOR

PRELIMINARY

INTRODUCTION

FEATURES

- Two enhanced CMOS R6502 CPU's in one device
  - —Common memory and I/O
  - —Shared data and subroutines
  - —Independent CPU registers and interrupt vectors
  - —Independent reset operation and programs
  - —R6502 software and timing compatible
- 10 new instructions for faster and smaller programs
  - —Unsigned Multiply (MUL)
  - —Set and Reset Memory Bit (SMB and RMB)
  - —Branch on Bit Set and Reset (BBS and BBR)
  - —Unconditional Branch (BRA)
  - —Push and Pull Index Registers (PHX, PHY, PLX, PLY)
- Microcomputer/microprocessor/peripheral controller operation
  - —Stand-alone microcomputer
    2048 × 8 mask programmable ROM
    128 × 8 random access memory (RAM)
  - —Enhanced microprocessor
    Built-in RAM, ROM and I/O with expandability
    8-, 12- or 16-bit extension address bus
  - —Programmable peripheral controller
    Host data bus interface (Z80/8080 or 6500/6800 option)
    Self-contained or expandable
- 16-bit Counter/Timer A with eight modes, and prescaler
  - —Timer Off
  - —Free-Run Event Counter
  - —Free-Run Pulse Width Measurement
  - —One-Shot Retriggerable Timer
  - —One-Shot Interval Timer
  - —Free-Run Interval Timer
  - —One-Shot Pulse Generator
  - —Free-Run Pulse Generator
- 16-bit Counter/Timer B with four modes
  - —Free-Run Interval Timer
  - —Free-Run Pulse Generator
  - —Event Counter
  - —Pulse Width Measurement
- Up to 52 general purpose input/output lines
  - —Five bidirectional 8-bit ports (PA, PB, PC, PD and PF)
  - —One 8-bit output port (PE)
  - —One 4-bit input port (PG)
  - —Multi-purpose operation for selected ports
- Nine interrupts
  - —Positive and negative edge detect
  - —Low level detect (external IRQ)
  - —Counter/Timer A and B underflow
  - —Inter-processor communication
  - —Host computer data transfer
  - —Non-maskable
  - —Reset
- Flexible system operation
  - —Memory mapped I/O for easy programming
  - —Page zero location for memory efficient access
- Low power at normal frequency (40 mw at 2 MHz)
- Reduced power at low frequency (2.0 mw at 2 MHz/128)
- System clock rates from 10 KHz to 4 MHz
- 5V ± 10% power supply
- 64-pin QUIP

SUMMARY

The Rockwell R65C00/21 is a complete, high performance 8-bit, CMOS dual microcomputer in a single chip and is compatible with all R6500 microprocessors except that it has additional instructions including a 10-clock time multiply.

The R65C00/21 consists of two enhanced instruction set 6502 CPU's in one device. The device also has 2048 bytes of Read-Only Memory (ROM), 128 bytes of Random Access Memory (RAM) and versatile interface circuitry. The interface circuitry consists of two multimode programmable 16-bit counter/timers and 52 general purpose input/output lines. Some of these input/output lines may be used as address, data and control lines for expanded systems or as data and control lines when the R65C00/21 is used as a programmable peripheral controller.

The two CPU's in the R65C00/21 are functionally independent. Each has its own set of registers, its own reset and interrupt vectors and operates under control of its own program. The two CPU's do, however, share the same memory and system I/O resources. This allows direct communication between the two CPU's and allows sharing of subroutines and common data areas where desired. Programming and system design for applications which require simultaneous control of two or more independent asynchronous processes is thus simplified because one CPU may control one process while the other controls another one. Consequently, complex programming usually needed to interleave the control functions or to implement an interrupt driven system, is not required.

In a multiple computer approach, both processors may need the same subroutines so that some portions of memory must be duplicated in both systems. The dual CPU's share the same program memory, therefore only one set of subroutines is required and both CPU's may even be using them at the same time without interference.

In addition to the dual CPU's, the R65C00/21 also has the innovative architecture and the demonstrated high performance of the well established R6502 CPU, flexible input/output which provides improvements over the R6522 Versatile Interface Adapter (VIA) device, and production efficient on-chip ROM and RAM. These features make the R65C00/21 a leading candidate for most imbedded microcomputer applications.

A system using the R65C00/21 Dual CMOS Microcomputer will be simpler in design, use less program memory, require fewer components, reduce circuit board sizes, simplify test requirements, and minimize field maintenance—all contributing to lower production and support costs. In addition, simpler designs shorten development effort and time—leading to reduced development costs and faster product to market.

The R65C29 Dual CMOS Microprocessor, a ROM-less version of the R65C00/21 with permanently extended data and address bus, is also available. The R65C29 is ideal for dual CPU applications requiring changeable ROM and/or extended RAM, ROM or I/O, and can also be used for R65C00/21 prototype circuit development. The R65C00/21 can also operate in an emulation mode, like the R65C29, with its internal ROM disabled.

DEVELOPMENT SYSTEM SUPPORT

Prototype circuit and software development support are available using the Rockwell Design Center (RDC) and R65C00/21 Personality Module. Program development and debugging aids such as text editing, symbolic assembly with conditionals and macros at the absolute and relocatable/linking level, and single/multiple step execution with instruction/data tracing are provided. Real-time in-circuit emulation in the target environment is also supported.

NOTE

All descriptions of R65C00/21 operation in this document also apply to the R65C29 except for internal ROM, and as otherwise noted.

ORDERING INFORMATION

The R65C00/21 Dual CMOS Microcomputer can be ordered in volume quantities with the following speed capability and mask option indicated in the R65C00/21 ROM Code Order Form (Document Order No. 2134)

- 1, 2, 3, or 4 MHz system clock (Ø2)
- Crystal/master clock or slaved clock input mask option

The R65C29 Dual CMOS Microprocessor has the following characteristics:

- Crystal/master clock input
- 8-bit data bus and 16-bit address bus extension
- No internal ROM

INTERFACE

The interfaces for the R65C00/21 and R65C29 are illustrated in Figure 1.

The pin assignments for the R65C00/21 and the R65C29 are shown in Figure 2. The R65C29 pin assignments are the same as the R65C00/21 except that bus expansion functions are permanently assigned instead of general purpose ports D and E.

The interface signals for the R65C00/21 and R65C29 are described in Table 2. The descriptions of the selectable bus expansion pins (16-bit address mode) for the R65C00/21 apply to permanent bus expansion pins for the R65C29.

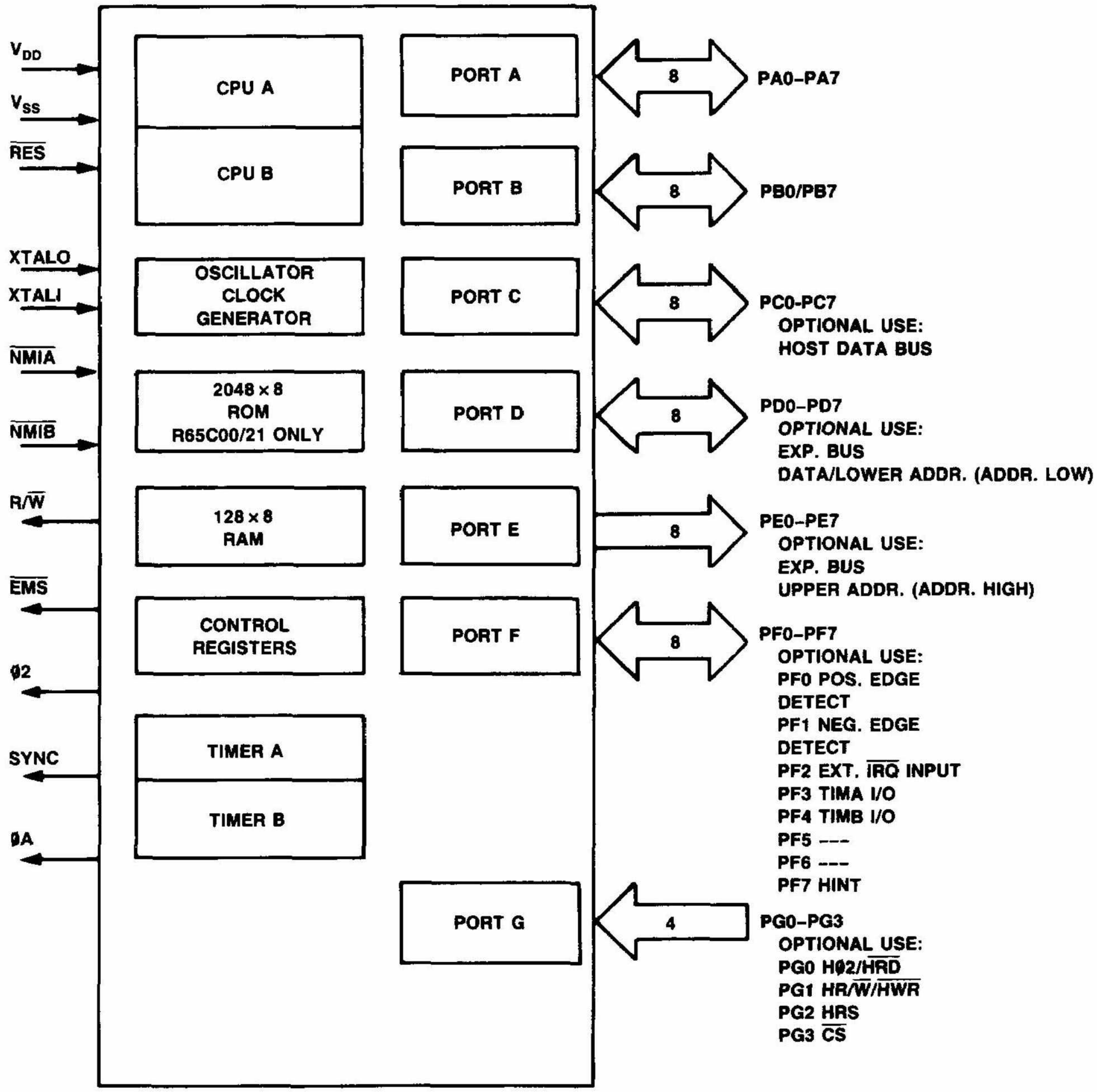


Figure 1. R65C00/21 and R65C29 Interface Diagram

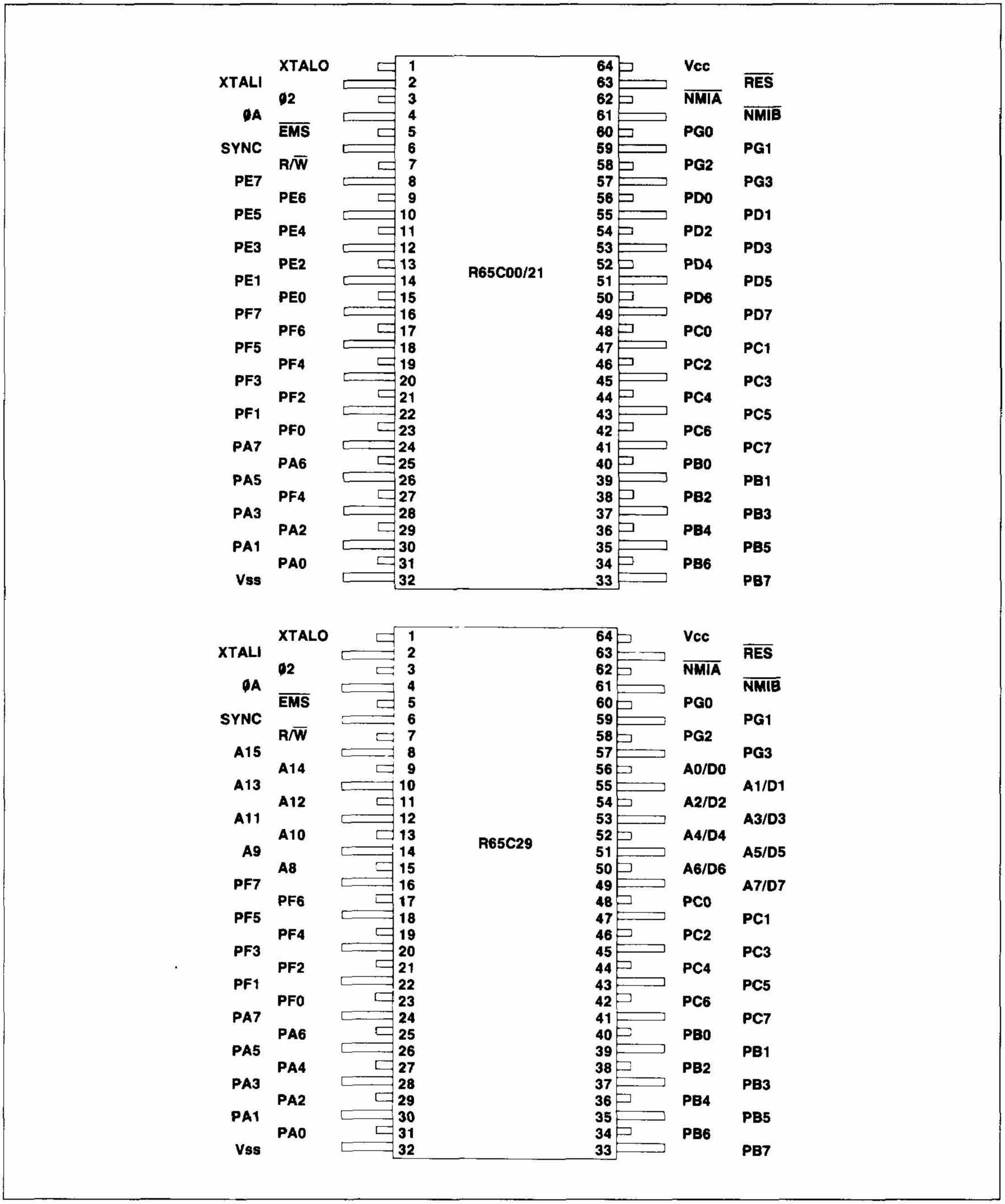


Figure 2. R65C00/21 and R65C29 Pin Assignments

Table 1. R65C00/21 Pin Description

Signal Name	Pin No.	I/O	Description
PA0-PA7	31-24	I/O	**Port A.** General purpose 8-bit I/O Port A.
PB0-PB7	40-33	I/O	**Port B.** General purpose 8-bit I/O Port B.
PC0-PC7	48-41	I/O	**Port C.** General purpose 8-bit I/O Port C. Host Data Bus in Host Mode.
PD0-PD7	56-49	I/O	**Port D.** General purpose 8-bit I/O Port D. Multiplexed lower address (A0 to A7) and Data Bus (D0-D7) when Bus Extension Mode is selected.
PE0-PE7	15-8	O	**Port E.** General purpose 8-bit output Port E. Upper address (A8 to A11 or A8 to A15) when Bus Expansion Mode is selected.
PF0-PF7	23-16	I/O	**Port F.** General purpose 8-bit I/O Port F. Under software control, each line has alternate functions as follows:
PF0NEG (PF0)	23	I	**PF0 Positive Edge Detect.** Maskable CPU interrupt on PF0 Positive Transition.
PF1POS (PF)	22	I	**PF1 Negative Edge Detect.** Maskable CPU interrupt on PF1 Negative Transition.
PF2LOW (PF2)	21	I	**PF2 Low Level Detect.** Maskable CPU interrupt on PF2 Low (external IRQ).
TIMA (PF3)	20	I/O	**Timer A External Input/Output.**
TIMB (PF4)	19	I/O	**Timer B External Input/Output.**
$\overline{\text{HINT}}$ (PF7)	16	O	**Host Interrupt.** Active-low maskable interrupt request to Host.
PG0-PG3	60-57	I	**Port G.** General purpose 4-bit input Port G. Under software control, Port G serves as the Host Control Bus as follows:
HØ2/$\overline{\text{HRD}}$ (PG0)	60	I	**Host Bus Clock/Read Strobe Input.** Ø2 for 6500/6800 bus; Read Strobe for Z80/8080 bus.
HR/$\overline{\text{W}}$/$\overline{\text{HWR}}$ (PG1)	59	I	**Host Bus Read-Write/Write Strobe Input.** R/$\overline{\text{W}}$ for 6500/6800 bus; Write Strobe for Z80/8080 bus.
HRS (PG2)	58	I	**Host Bus Register Select Input.** Low selects Data Buffer; high selects Status Flags.
$\overline{\text{CS}}$ (PG3)	57	I	**Host Bus Active-Low Chip Select Input.** Low selects Host Bus operation depending on HRS and HR/$\overline{\text{W}}$/$\overline{\text{HWR}}$ coding and Host Control and Status Register contents; high disables Host Bus interface.
$\overline{\text{RES}}$	63	I	**Reset.** Active-low Reset input initializes R65C00/21 to initial conditions—resets all registers and I/O lines.
$\overline{\text{NMIA}}$	62	I	**CPU A Non-Maskable Interrupt.** Non-maskable negative edge sensitive interrupt input to CPU A.
$\overline{\text{NMIB}}$	61	I	**CPU B Non-Maskable Interrupt.** Non-maskable negative edge sensitive interrupt input to CPU B.
$\overline{\text{EMS}}$	5	O	**External Memory Strobe.** Active-low.
Ø2	3	O	**System Phase 2 Clock Output.** Maskable as system clock input for slave operation.
R/$\overline{\text{W}}$	7	O	**Read/Write.** Read/write control output. High during read, low during write.
SYNC	6	O	**Sync.** Instruction sync output. High When Op Code fetched
ØA	4	O	**Phase A.** Phase A clock output. High during CPU A bus cycle, low during CPU B bus cycle.
XTALO	1	O	**Crystal/Master Clock Return.** Output connection to crystal (or no connection if external master clock connected to XTALI). Input frequency is two times system clock (Ø2) rate.
XTALI	2	I	**Crystal/Master Clock Input.** Input connection from crystal (or external master clock).
VCC	64		**Power.** 5.0 Vdc.
VSS	32		**GND.** Signal and power ground.

FUNCTIONAL DESCRIPTION

The R65C00/21 consists of two central processor units (CPU's), a 2048 × 8 read-only memory (ROM), a 128 × 8 random access memory (RAM), five 8-bit parallel I/O ports, one 8-bit output port, one 4-bit input port, two 16-bit counter/timer systems, a variety of I/O control registers, and an independent interrupt control system for each CPU. All of the ROM, RAM, I/O, internal buses, and the arithmetic logic unit (ALU) are shared by the two CPU's. A memory map of the system is shown in Figure 3. An overall block diagram of the R65C00/21 is shown in Figure 4.

NOTE

Throughout this document, unless specified otherwise, all memory or register address locations are specified in hexadecimal notation.

INTERNAL MEMORY

INTERNAL READ-ONLY-MEMORY (ROM)

The ROM in the R65C00/21 consists of 2048 (2K) bytes of mask programmable memory with an address space from F800 to FFFF. ROM locations FFF2 through FFFF are assigned to interrupt and reset vectors for the two CPU's.

INTERNAL RANDOM ACCESS MEMORY (RAM)

The internal RAM consists of 128 bytes of read/write memory with assigned page zero addresses of 0080 through 00FF.

EXTERNAL MEMORY

External memory can be addressed by selecting the Bus Expansion Mode in the Bus Control Register. Address space from 0200 through EFFF may be accessed for either RAM, ROM, or I/O purposes as the particular application requires it. In addition, there are 32 bytes from 0020 through 003F which may be used for I/O expansion and 256 bytes from 0100 through 01FF which may be external RAM.

CPU LOGIC

Each CPU in the R65C00/21 is effectively a standard R6502 CPU with 10 extra instructions utilizing 40 operation codes which are unused in the R6502. Therefore, each CPU has an 8-bit accumulator, two 8-bit index registers (X and Y), an 8-bit Stack Pointer Register, an 8-bit Status Register, a 16-bit Program Counter, independent interrupt circuitry, and an instruction register with state counter. The internal buses, memory, instruction decoding circuitry, and ALU are shared by the two CPU's on alternate clock cycles.

ACCUMULATORS

The accumulator in each CPU is a general purpose 8-bit register that stores the results of most arithmetic and logic operations. Additionally, the accumulator contains one of the two data words used in these operations.

INDEX REGISTERS

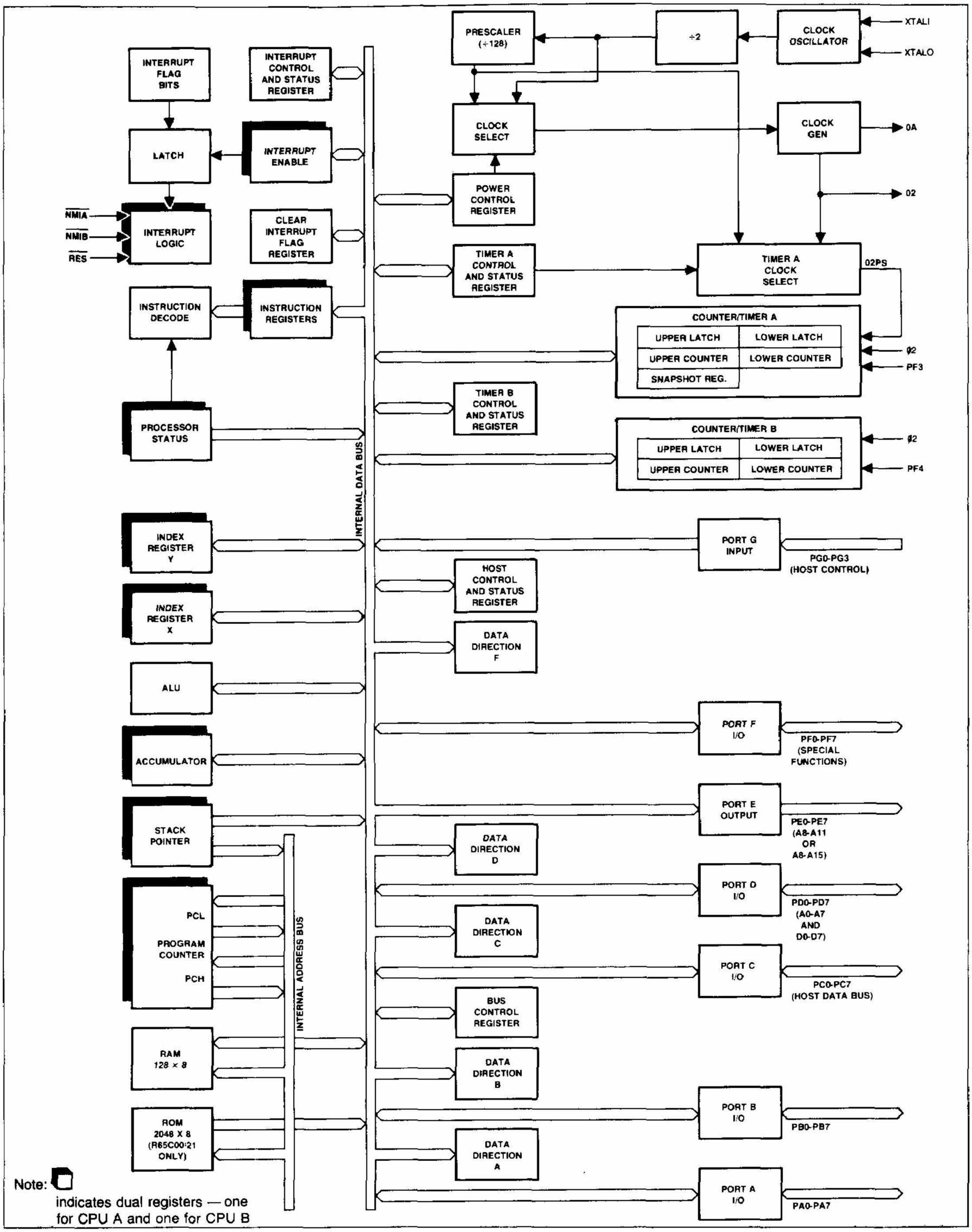
Each CPU has two index registers, X and Y. Each index register may be used as a modifier to a base address supplied as a part of the instruction being processed. The resulting effective address is usually the sum of the base address plus the contents of the indicated index register. The index registers are used in a number of the addressing modes including zero page indexed, absolute indexed, post-indexed indirect and pre-indexed indirect. Each index register also has a family of instructions which allow loading, storing, incrementing, decrementing, and comparing the contents of the register. These are discussed thoroughly in the R6500 Programming Manual (Order No. 202).

ADDRESS (HEX)	
0000 001F	I/O AND CONTROL REGISTERS
0020 003F	EXTERNAL I/O EXPANSION[2]
0040 007F	NOT ACCESSIBLE
0080 00FF	INTERNAL RAM (128 BYTES)[1] (SHARED WITH 0180-01FF)
0100 017F	EXTERNAL RAM EXPANSION[2]
0180 01FF	INTERNAL RAM (128 BYTES)[1] (SHARED WITH 0080-00FF)
0200 EFFF	EXTERNAL MEMORY AND I/O EXPANSION[2]
F000 F7FF	NOT ACCESSIBLE
F800 FFF1	INTERNAL ROM (2048 BYTES)
FFF2 FFF3	NMIB VECTOR
FFF4 FFF5	RESB VECTOR
FFF6 FFF7	IRQB VECTOR
FFF8 FFF9	NOT USED
FFFA FFFB	NMIA VECTOR
FFFC FFFD	RESA VECTOR
FFFE FFFF	IRQA VECTOR

Notes:

1. When bit 4 of the Bus Control Register (BCR) is a 0 (default value), the 128 bytes of internal RAM are redundantly mapped into both page zero and page one and are addressable as either 0080-00FF or 0180-01FF. When BCR bit 4 is a 1, all of page one RAM (256 bytes) is mapped externally (0100-01FF) and the 128 bytes of internal RAM are dedicated to page zero (0080-00FF).
2. Accessible in bus expansion mode.

Figure 3. R65C00/21 Memory Map



Note: ▣ indicates dual registers — one for CPU A and one for CPU B

Figure 4. R65C00/21 and R65C29 Block Diagram

STACK POINTERS

Each CPU in the R65C00/21 has its own independent 8-bit Stack Pointer Register located in RAM on page zero/one and is pointed to by a Stack Pointer. Each Stack Register is automatically incremented and decremented under control of the appropriate CPU to perform proper stack manipulations in response to user instructions, an IRQ interrupt or an external NMI interrupt of the appropriate CPU. The Stack Pointers must be initialized by the user program.

These stacks allow simple implementation of multiple level independent interrupts in each CPU, subroutine nesting, and simplification of many types of data manipulation without the programmer continually being aware of specific memory addresses. The JSR, BRK, RTI, RTS, PHA, PLA, PHP, PLP, PHX, PLX, PHY and PLY instructions all make use of the stack and the appropriate CPU's Stack Pointer.

Each stack may be visualized as a deck of cards which may only be accessed from the bottom of the deck. The value to be stored is written on a card and then that card is placed on the bottom of the deck (pushed onto the stack). When the data are to be read, the bottom card is removed from the deck and the value on it transferred to the appropriate register (pulled from the stack to the specific register). Each time data are to be used as an address, the value is stored in the addressed memory cell, and the Stack Pointer is decremented by 1. When the data are read (or "pulled") from the stack, the Stack Pointer is incremented by 1 and the resulting value can be used to address the data. The data are read from the addressed memory cell and then transferred to the appropriate register in the CPU.

Each CPU must have an independent starting location for its stack. It is the programmer's responsibility to see that the RAM utilized for each CPU stack does not conflict. It is recommended that the CPU requiring less depth in its stack be assigned the OXFF location and the other stack be started a safe distance below it. The two stacks are physically located either on page zero (although addressed as page one) for single-chip operation, or externally on page one when extended addressing is selected. (See Note 1 in Figure 3). The default areas for the stacks are on page zero. In either case, both stacks are on the same page.

ARITHMETIC AND LOGIC UNIT (ALU)

All arithmetic and logic operations for both CPU's take place in a shared ALU. Incrementing and decrementing of the index registers and memory also take place here. The ALU stores data for only one cycle. Consequently, data placed on the inputs at the beginning of a cycle are processed and gated to one of the registers, or to memory, during the next cycle.

Each bit of the ALU has two inputs. These inputs may be tied to various internal buses or to a logic zero; the ALU then generates the function (AND, OR, SUM, etc.) using the data on the two inputs.

PROGRAM COUNTERS

The 16-bit program counters for each CPU provide addresses that step each processor through sequential instructions in a stored program. The program counter for each CPU is initially set to the value stored as the reset vector in CPU A (RESA at FFFC) and in CPU B (RESB at FFF4) when power is applied to the R65C00/21. Each time a processor fetches an instruction from memory, the lower (least significant) byte of its program counter (PCL) is placed on the low-order eight bits of the address bus and the higher (most significant) byte of the program counter (PCH) is placed on the high-order eight bits of the address bus. The counter is incremented each time an instruction or operand is fetched from memory.

The contents of the program counter are replaced with a new value when a JMP, JSR, RTS, RTI, BRK, or any of the branch instructions are executed. Also, the program counter value is replaced when an external non-maskable interrupt $\overline{\text{NMIA}}$ or $\overline{\text{NMIB}}$, an internal interrupt request, an external interrupt request via PF2 (see Port F description) or reset ($\overline{\text{RES}}$) occurs.

INSTRUCTION REGISTERS AND INSTRUCTION DECODE

Instructions selected by the program counter are fetched from ROM or RAM (or Port D if in Expanded Bus Mode) and gated onto the internal data bus. These instructions are latched into the proper instruction register and then decoded using common decoding circuits for both CPU's. Timing, status bits, and interrupt controls are interpreted together with the instruction code to generate control signals for the various registers in the appropriate CPU.

INTERRUPT LOGIC

Each CPU has its own logic which controls the sequencing of three types of interrupts: $\overline{\text{RES}}$, $\overline{\text{NMI}}$, and $\overline{\text{IRQ}}$. The same RESET ($\overline{\text{RES}}$) pin is used for both CPU's; consequently, reset occurs on both CPU's at the same time. A different reset vector (RESA and RESB) exists for each CPU to allow initialization of the separate and independent programs.

Separate pins are used for the two processors' non-maskable interrupts ($\overline{\text{NMIA}}$ and $\overline{\text{NMIB}}$). Each processor has its own NMI vector; CPU A uses NMIA Vector at FFFA and CPU B uses NMIB Vector at FFF2.

Three different types of external interrupt conditions can be detected by connecting the external signal to one of three Port F input pins. A positive-going edge, a negative-going edge, and an external interrupt request ($\overline{\text{IRQ}}$), i.e., a low level, can be detected on PF0, PF1 and PF2, respectively. Internally, IRQ conditions can be generated by time-out of either of the two 16-bit counter/timers, upon interprocessor-communication request by the other CPU, or by the Host Interface Port.

In each case, the interrupt condition is reported as an interrupt flag in a control/status register associated with the functional area. Each CPU can either enable or disable IRQ generation by setting or resetting a corresponding interrupt enable bit in the same or associated control/status register.

Furthermore, each CPU can control whether or not its processing is interrupted when an interrupt request ($\overline{\text{IRQ}}$) is generated. Each CPU has its own Processor Status Register (PSRA and PSRB) with the capability of disabling $\overline{\text{IRQ}}$ interrupts when its own "I flag" bit is a 1.

NEW AND MODIFIED INSTRUCTIONS

In addition to the standard R6502 instruction set, ten new instructions have been added and minor timing and other changes have been made to a few other instructions. All of these additions and changes are discussed in this section. Refer to the Instruction Set Op Code Matrix for the operation codes and addressing modes of all instructions. The times indicated for each instruction are given in terms of CPU clock-times.

UNSIGNED MULTIPLY (MUL)

The 10 clock-time hardware multiply instruction multiplies the 8-bit contents of the Y register by the 8-bit contents of the A register to give a 16-bit product. At the completion of the multiply operation, the most significant half of the product resides in the A register and the least significant half in the Y register. This operation uses unsigned numbers only. This instruction uses the implied addressing mode and, consequently, requires one byte for the op code.

SET MEMORY BIT (SMB m, ADDRESS.)

This instruction uses zero page addressing only and requires five cycle times. It sets the designated bit in the addressed memory cell or I/O port to a 1. The first byte of the two-byte instruction identifies the operation and the bit to be set while the second byte designates the address of the word in which the bit is to be set. Eight op codes are used for the eight bit locations in a byte.

RESET MEMORY BIT (RMB m, ADDRESS.)

This instruction operates in the same way as the SMB instruction except that the bit is set to 0.

BRANCH ON BIT SET RELATIVE (BBS m, ADDRESS, DESTINATION)

This instruction tests one of eight bits designated by a three-bit immediate field within the first byte of the instruction. The second byte designates the address of the byte to be tested within the zero page address range (memory or I/O ports). The third byte of the instruction specifies the 8-bit relative address to which the instruction branches if the bit tested is a 1. If the bit tested is not set to a 1, the next sequential instruction is executed. This instruction requires five cycles if the branch is not executed, six cycles if the branch executes to the same page, or seven cycles if it branches to a different page.

BRANCH ON BIT RESET RELATIVE (BBR m, ADDRESS, DESTINATION)

This instruction is similar to the BBS instruction except that the branch takes place if the bit tested is a 0.

INDEX REGISTER STACK OPERATIONS (PHX, PLX, PHY, AND PLY)

These instructions are similar to the PHA and PLA instructions in the conventional R6502 except that they push or pull the X or Y registers to and from the stack, respectively. The push instructions require three instruction cycles and the pull instructions require four cycles.

UNCONDITIONAL BRANCH (BRA)

This unconditional branch is a branch always instruction. It operates similar to the conditional branches of the R6502 except that the relative branch always occurs. It executes in three cycles if the branch is to the same page or four cycles if it is not. Two bytes are required, one for the op code and the other for the relative address.

INSTRUCTION DIFFERENCES FROM R6502

Decimal add and decimal subtract instructions on the R65C00/21 require one cycle time longer than their binary equivalents. The add and subtract times are the same for both decimal and binary operation on the R6502.

The decimal mode flag (D) in the processor status registers default to binary (D=0) operation when the R65C00/21 is RESET, whereas this bit is uninitialized on the R6502.

The indirect jump instruction increments the page address when the indirect pointer crosses a page boundary, whereas on the R6502 it does not.

PROCESSOR STATUS REGISTERS

Each CPU has its own 8-bit Processor Status Register. Each register contains seven status flags. Some of these flags are controlled by the user program; others may be controlled both by the user's program and the appropriate CPU. The R65C00/21 instruction set contains a number of conditional branch instructions which are designed to allow testing of these flags.

CARRY BIT (C)

The carry bit (C) can be considered the ninth bit of an arithmetic operation. It is set to a 1 if a carry from the eighth bit has occurred, or it is cleared to 0 if no carry has occurred, as a result of arithmetic or shift operations.

The carry bit may be set or cleared under program control by use of the Set Carry (SEC) or Clear Carry (CLC) instructions, respectively. Other operations which affect the carry bit are ADC, ASL, CMP, CPX, CPY, LSR, PLP, ROL, ROR, RTI, and SBC.

ZERO BIT (Z)

The zero bit (Z) is set to a 1 by the CPU during any data movements, or calculations, which sets all eight bits of the result to zero for that CPU. This bit is cleared to a 0 when all eight bits of a data movement, or calculation, operations are not all zero for that CPU. The R6500 instruction set contains no instruction to specifically set or clear the Z flag bit. The Z flag bit is, however, affected by the following instructions: ADC, AND, ASL, BIT, CMP, CPX, CPY, DEC, DEX, DEY, EOR, INC, INX, INY, LDA, LDX, LSR, ORA, PLA, PLP, PLX, PLY, ROL, ROR, RTI, SBC, TAX, TAY, TXA, TSX, and TYA.

INTERRUPT DISABLE BIT (I)

The interrupt disable bit (I) controls the servicing of an interrupt request ($\overline{\text{IRQ}}$). If the I bit is set to a 0 in the Processor Status Register of one, or both, of the CPU's, the $\overline{\text{IRQ}}$ signal will be serviced by that particular CPU. If the bit is set to a 1 for one or both of the CPU's, the $\overline{\text{IRQ}}$ signal will be ignored by that CPU. Each CPU will set its interrupt disable bit to a 1 if a $\overline{\text{RES}}$, an $\overline{\text{IRQ}}$, or its non-maskable interrupt ($\overline{\text{NMI}}$) signal is detected. Interrupting one processor does not affect the other one unless it is programmed to respond to the same interrupt.

The I bit is cleared for each CPU when that CPU executes a Clear Interrupt Disable (CLI) instruction and is set under software control by a Set Interrupt Disable (SEI) instruction. This bit is also set by the Break (BRK) instruction. The Return From Interrupt (RTI) and Pull Processor Status (PLP) instructions also affect the I bit by setting it to the value which was stored on the stack.

DECIMAL MODE BIT (D)

The decimal mode bit (D) controls the arithmetic mode of its CPU. When this bit is set to a 1, the adder operates as a decimal adder for the Add with Carry (ADC) and the Subtract With Carry (SBC) instructions. These instructions, in the decimal mode, require one additional CPU cycle time compared with binary mode or the decimal mode in the conventional R6500. (In the conventional R6500, the decimal and binary arithmetic operations are the same speed.) When the bit is a 0, the arithmetic is performed in straight binary. The decimal mode is controlled only by the programmer for each of the CPU's. The Set Decimal Mode (SED) instruction causes decimal arithmetic to be performed and the Clear Decimal Mode (CLD) instruction causes binary arithmetic to be performed by that CPU. The PLP and RTI instructions also affect the decimal mode bit.

The D bit for each CPU is automatically set to the zero state (binary mode) when the R65C00/21 is reset by $\overline{\text{RES}}$.

BREAK BIT (B)

The break bit (B) determines the type of condition which caused the IRQ service routine to be entered. If the IRQ service routine was entered because a BRK instruction was executed by its CPU, the B bit is set to a 1. If the service routine was entered because of an $\overline{\text{IRQ}}$ signal being generated, the B is set to a 0. There are no instructions which directly set or clear this bit.

OVERFLOW BIT (V)

The overflow bit (V) indicates that the result of a signed binary addition or subtraction operation is a value which cannot be contained in seven bits (outside the range of −128 to +127). This indicator only has meaning when signed arithmetic is performed. In this case, the arithmetic operations are being performed on the sign and seven magnitude bits for one byte, or the most significant byte of a longer signed number. When the ADC or SBC instruction is executed, the overflow bit is set to a 1 if the polarity of the sign bit is changed because the result exceeds +127 or −128 in absolute magnitude. Otherwise, the V bit is cleared to a 0. The V bit may be cleared by the programmer by executing a Clear Overflow (CLV) instruction in the appropriate CPU.

The overflow bit is also affected by the BIT instruction. The BIT instruction samples specific bits in memory or I/O interrupt status words. Most of the I/O devices used in the R6500 family and most of the interrupt flags in the R65C00/21 have interrupt flags in the upper two bits of the register. The BIT command copies these two most significant bits of the addressed word into the N and V flags. The V flag is set to the same state as bit 6 of the addressed words and the N flag copies bit 7.

The instructions which affect the V flag are ADC, BIT, CLV, PLP, RTI and SBC.

NEGATIVE BIT (N)

The negative bit (N) indicates that the sign bit (bit 7) in the resulting value of a data movement or arithmetic operation is a 1. If the value represents a signed number, the most significant bit being a 1 indicates a negative number. If the sign bit is a 0, the result is interpreted as a positive value. The BIT instruction copies the most significant bit of the addressed memory cell or I/O register into the N flag bit.

There are no instructions that set or clear the N bit directly since the N bit represents only the status of a result. The instructions which produce a result that affects the state of the N bit are AND, ASL, BIT, CMP, CPX, CPY, DEC, DEX, DEY, EOR, INC, INX, INY, LDA, LDX, LDY, LSR, ORA, PLA, PLP, PLX, PLY, ROL, ROR, RTI, SBC, TAX, TAY, TSX, TXA, and TYA.

Processor Status Registers (PSRA and PSRB)

7	6	5	4	3	2	1	0
NEG (N)	OVFL (V)	NOT USED	BRK (B)	DEC (D)	IRQ ENBL (I)	ZERO (Z)	CARRY (C)

Bit 7	**Negative (N)**[1]
1	Negative Value
0	Positive Value
Bit 6	**Overflow (0)**[1]
1	Overflow Set
0	Overflow Clear
Bit 5	**Not Used** (Don't care value)
Bit 4	**Break Command (B)**[1]
1	Break Command
0	Non-break Command
Bit 3	**Decimal Mode (D)**[3]
1	Decimal Mode
0	Binary Mode
Bit 2	**Interrupt Enable (I)**[2]
1	IRQ Interrupt Disable
0	IRQ Interrupt Enable
Bit 1	**Zero (Z)**[1]
1	Zero Result
0	Non-Zero Result
Bit 0	**Carry (C)**[1]
1	Carry Set
0	Carry Clear

Notes:
1. Not initialized by $\overline{\text{RES}}$.
2. Set logic 1 by $\overline{\text{RES}}$.
3. Cleared to logic 0 by $\overline{\text{RES}}$.
4. There are two Processor Status Registers, one for each CPU.

INPUT/OUTPUT AND CONTROL/STATUS REGISTERS

REGISTER ADDRESSES

Table 2 shows the input/output, control/status and timer/counter registers which are addressed on page zero from locations 00 through 1D. Some of the registers combine other functions when they are read or written. The table lists both the primary and secondary types of functions. Table 3 summarizes the register formats.

All control/status registers and data direction registers are cleared to zero by a $\overline{\text{RES}}$. Thus, the zero state of each bit defines the default operating modes. Each register is associated with a functional area in the microcomputer, e.g., parallel input/output, timer/counter, bus control, etc. The detail operation of each register is defined in the appropriate sections.

Thirteen registers are used for input/output functions and nine registers used for timer/counter functions. The use of these registers is discussed in later sections.

Seven control/status registers control and monitor the basic operation of the R65C00/21. The registers and their primary functions are as follows:

BCR Bus Control Register—defines expansion bus modes
HCSR Host Control and Status Register—defines host bus and interrupts
ICSR Interrupt Control and Status Register—enables and reports interrupt conditions
CIFR Clear Interrupt Flags Register
PCR Power Control Register—selects low power mode
TACSR Timer A Control and Status Register—controls and monitors Timer A operation
TBCSR Timer B Control and Status Register—controls and monitors Timer B operation

Table 2. I/O, Control/Status and Timer Registers

Address	Read	Write	Register Names/Notes
00	PA Data	PA Data	Port A Data I/O
01	PB Data	PB Data	Port B Data I/O
02	PC Data, 0→IBF[3]	PC Data, 1→OBE, 0→RSO[3]	Port C Data I/O
03	PC Data, 0→IBF[3]	PC Data, 1→OBE, 1→RSO[3]	Port C Data I/O
04	PD Data[1]	PD Data[1]	Port D Data I/O
05	—	PE Data[1]	Port E Data Output Only
06	PF Data	PF Data	Port F Data I/O
07	PG Data	—	Port G Data Input Only
08	—	PA Direction	Port A Direction
09	—	PB Direction	Port B Direction
0A	—	PC Direction	Port C Direction
0B	—	PC Direction	Port C Direction
0C	—	PD Direction[1]	Port D Direction
0D	—	—	
0E	—	PF Direction	Port F Direction
0F	—	—	
10	BCR	BCR	Bus Control Register
11	HCSR	HCSR	Host Control and Status Register
12	ICSR	ICSR	Interrupt Control and Status Register
13	—	CIFR	Clear Interrupt Flags Register
14	—	IPCIR	Inter-Processor Communication Interrupt Register
15	PCR	PCR	Power Control Register
16	TACSR	TACSR	Timer A Control and Status Register
17	LCA, UCA→SLA	LLA	Timer A Lower Counter (LCA)/Lower Latch (LLA)
18	SLA	ULA	Timer A Snapshot Latch (SLA)/Upper Latch (ULA)
19	SLA, 0→UFA[2]	ULA, ULA→UCA, LLA→LCA, 0→UFA[2]	Timer A Snapshot Latch (SLA)/Upper Latch, Download and Start Timer
1A	TBCSR	TBCSR	Timer B Control and Status Register
1B	LCB	LLB	Timer B Lower Counter (LCB)/Lower Latch (LLB)
1C	UCB	ULB	Timer B Upper Counter (UCB)/Upper Latch (ULB)
1D	UCB, 0→UFB[2]	ULB, ULB→UCB, LLB→LCB, 0→UFB[2]	Timer B Upper Counter (UCB)/Upper Latch (ULB), Download and Start Timer
1E	—	—	
1F	—	—	

Notes:
1. Addressed externally when in expanded bus mode.
2. Counter/Timer underflow flags:
   UFA = Timer A Underflow Flag bit in TACSR
   UFB = Timer B Underflow Flag bit in TBCSR
3. R65C00/21 to/from Host data transfer bits in HCSR:
   IBF = Input Buffer Full flag bit
   OBE = Output Buffer Empty flag bit
   RSI = Register Select Input bit
   RSO = Register Select Output bit
4. — = Not used—indeterminate data when read

Table 3. Control/Status Registers Formats Summary

Address (Hex)	Bit Number 7	6	5	4	3	2	1	0	
10	CPU A ACTIVE	NOT USED		PAGE ONE EXT	PORT A NIBBLE MODE		BUS EXTENSION MODE		BUS CONTROL REGISTER (BCR)
11	O/P BUFF FULL INT FLAG (OBF)	I/P BUFF FULL INT FLAG (IBF)	I/O REG SEL (RSI) (RSO)	NOT USED	I/OA INT ENBL I/OB INT ENBL	HOST INT ENBL	HOST BUS ENBL	HOST BUS TYPE	HOST CONTROL AND STATUS REGISTER (HCSR)
12	IPCA INT FLAG IPCB INT FLAG	PF2 LOW INT FLAG	PF1 NEG EDGE INT FLAG	PF0 POS EDGE INT FLAG	IPCA INT ENBL IPCB INT ENBL	PF2A INT ENBL IPCB INT ENBL	PF1A INT ENBL PF1B INT ENBL	PF0A INT ENBL PF0B INT ENBL	INTERRUPT CONTROL AND STATUS REGISTER (ICSR)
13	CLR IPCA INT FLAG CLR IPCB INT FLAG	NOT USED	CLR PF1 NEG INT FLAG	CLR PF0 POS INT FLAG	NOT USED				CLEAR INTERRUPT FLAGS REGISTER (CIFR)
14	WRITE ONLY REGISTER—NO SPECIFIC BIT (IPCIR)								INTER-PROCESSOR COMMUNICATION INTERRUPT REGISTER (IPCIR)
15	NOT USED						LOW PWR CPU B (LPB)	LOW PWR CPU A (LPA)	POWER CONTROL REGISTER (PCR)
16	TMR A UNFL FLAG (UFA)	PF3 LEVEL IND	NOT USED	TMR A INT ENBL	TMR A CLK PRESC SEL	TIMER A MODE SELECT			TIMER A CONTROL AND STATUS REGISTER (TACSR)
1A	TMR B UNFL FLAG (UFB)	PF4 LEVEL IND	NOT USED	TMR B INT ENBL	NOT USED		TIMER B MODE SELECT		TIMER B CONTROL AND STATUS REGISTER (TBCSR)

Note: All control and status registers are cleared to zero by $\overline{\text{RES}}$

INTERRUPT CONTROL AND STATUS

Unlike other R6500 family devices, the R65C00/21 does not concentrate the interrupt flags into a single register. The R65C00/21, in general, places the interrupt flags in registers which also have to do with the control of the particular function which can cause the interrupt.

Interrupt enable control is located in the following registers:

HCSR	Host Control and Status Register
ICSR	Interrupt Control and Status Register
TACSR	Timer A Control and Status Register
TBCSR	Timer B Control and Status Register

Portions of each of these registers relating to interrupt enables are duplicated for each of the two CPU's. However, only one memory address has been allocated so that each CPU uses the same address to select its own interrupt enables. The specific details of the usage of the interrupt control bits are discussed in the corresponding functional area.

CLOCK CIRCUITS

CLOCK OSCILLATOR

The internal clock oscillator generates the system clock (Ø2) which clocks all R65C00/21 operations. The system clock frequency ranges from 10 KHz to 4 MHz (the upper limit determined by the R65C00/21 part number) which is one-half the external crystal (or master clock) frequency. Each CPU in turn operates at one-half the system clock frequency (alternate cycles). All operations to memory or I/O take place at the system clock frequency. Since each CPU shares the common segments of the system on alternate system clock cycles, all internal operations occur at the system clock rate but, for CPU timing purposes, a CPU cycle rate of half the system rate is used. Thus with a 4 MHz crystal frequency, the system clock rate is 2 MHz and each CPU operates at an effective 1 MHz rate. Every two system clock periods sees one cycle devoted to CPU A and one cycle devoted to CPU B.

The Ø2 clock is normally routed externally to clock external memory operations in the extended bus mode. A mask option allows the Ø2 clock to be configured as an input so the R65C00/21 can operate in a slaved clock mode. In this case, the crystal input (XTALI) is grounded and crystal output (XTALO) is left open as shown in Figure 5.

LOW POWER OPERATION

The divide-by-128 clock prescaler operates in one of three ways (see Figure 6). One is the prescaler switched completely out which gives a system clock rate (Ø2) at one-half of the crystal frequency. Another way is to select the low power operation for both CPUs which switches in the clock prescaler. The clock prescaler divides the system clock frequency by 128 to generate the prescaled system clock rate (Ø2PS). This reduces the device power requirements and also reduces the counting rate of both counter/timers by a factor of 128. The third operating mode for the prescaler is to use it for prescaling Timer A only. This mode is discussed under the Counter/Timer Operation.

POWER CONTROL REGISTER (PCR)

Two bits in the Power Control Register (PCR) determine operation of the clock prescaler. Each CPU can set its own power control bit and read both of them. When both power control bits are a 1, the system switches to the low power operation at a clock rate of Ø2/128 (Ø2PS). The system reverts to normal power and speed when either power control bit is a 0 or when an enabled interrupt occurs. In the latter case, the system continues to operate at the low rate until the current instruction is completed, then it switches to the normal rate.

NOTE

An enabled interrupt automatically clears the PCR bit for the affected CPU. It must be set again by software to resume low power mode.

Power Control Register (PCR)

7	6	5	4	3	2	1	0
NOT USED						LOW PWR CPU B (LPB)	LOW PWR CPU A (LPA)

Bits 7-2	**Not Used** (Don't care)
Bit 1	**Low Power Mode Select for CPU B (LPB)**
1	Low power mode requested by CPU B
0	Normal power mode requested by CPU B
Bit 0	**Low Power Mode Select for CPU A (LPA)**
1	Low power mode requested by CPU A
0	Normal power mode requested by CPU A

Notes:
1. Both CPU's can read both bits.
2. Each CPU can only write its power control bit.
3. Both bits must be set to enable low power mode.

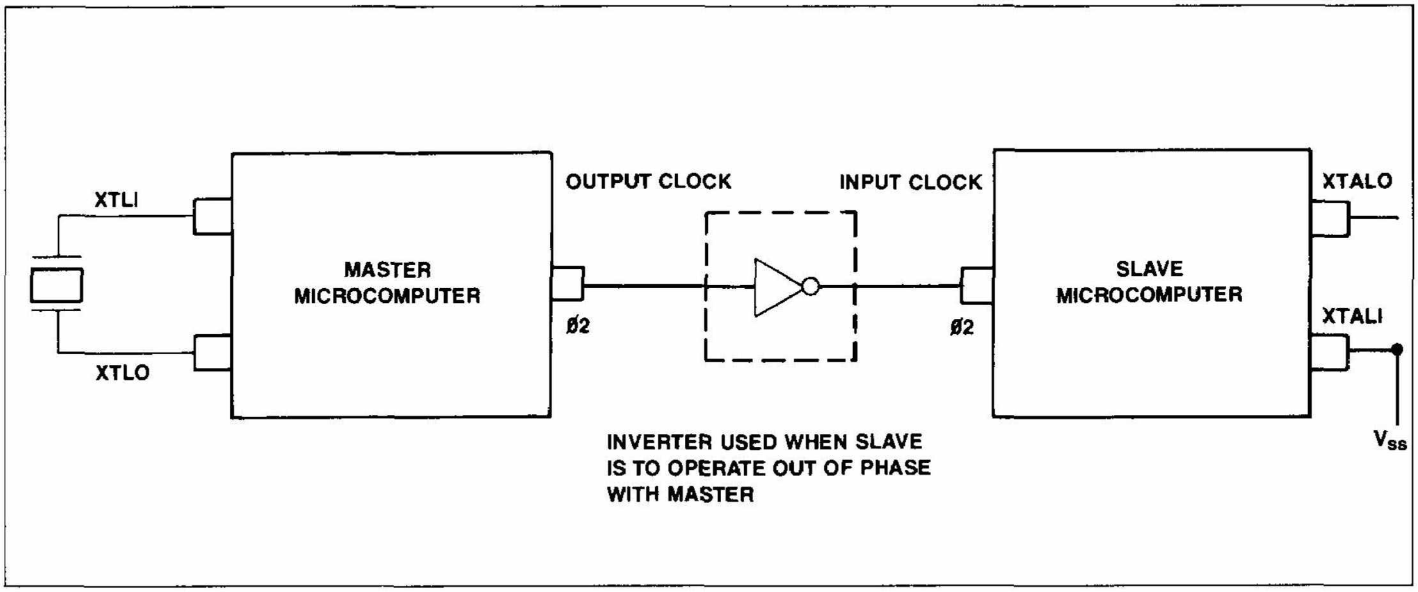
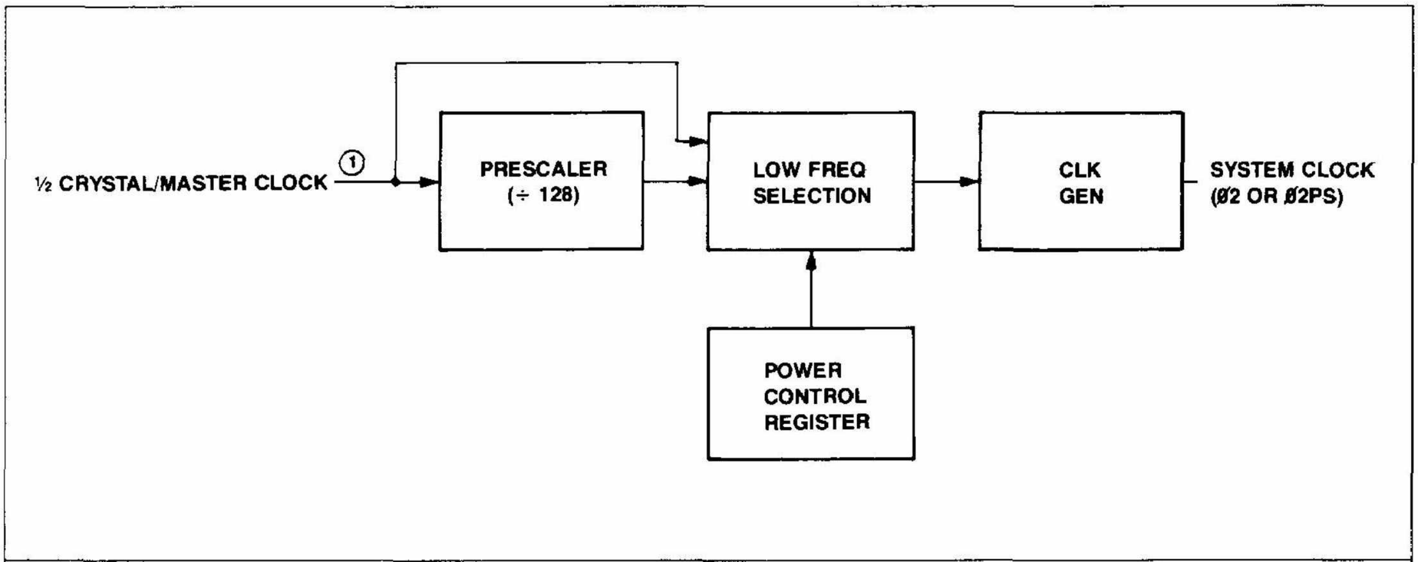


Figure 5. Master/Slave Cock Connection



Notes:

1. Crystal or Master clock frequency is divided by two internally.
2. System clock is Ø2 or Ø2PS (Ø2 ÷ 128) if low frequency operation for both CPU's is selected in the Power Control Register.
3. When a device is strapped for slave clock mode, the input Ø2 by-passes the prescaler (i.e., PCR bits will have no affect on the internal Ø2 clock rate of the slave processor.

Figure 6. System Clock Operation

PARALLEL INPUT/OUTPUT PORTS

The R65C00/21 parallel input/output interface consists of five 8-bit, bidirection input/output ports, one 8-bit output only port, and one 4-bit input only port.

BIDIRECTIONAL PORTS A, B, C, D AND F

The five 8-bit bidirectional ports (Ports A, B, C, D and F) each have an associated data direction register which configures individual data ports for either input or output. Port E is output only and port G is input only, therefore, no data direction registers are required for these two ports.

OUTPUT MODE

If the data direction register for a particular bit position in a bidirectional port is a 1, that bit is defined as an output pin. The information written into each bit position of the data word is loaded into a latch. The information will remain in that latch until new data is transmitted to the data word or the power is shut off. The output latches are individually connected to output drivers for each bit position for which a corresponding bit in the data direction register is a 1. The output drivers are double-ended, push-pull type. The drivers force the output pins high (≥2.4V) if the output data bit is a 1, or low (≥0.4V) if the output data bit is a 0. The output drivers are TTL compatible.

INPUT MODE

If the data direction register for a particular bit position in a bidirectional port is 0, that bit position is defined as an input pin. When the input/output port is read via an LDA, LDX, LDY, ADC, SBC, ORA, AND, EOR, or a BIT instruction, all of the information on that port's pins are read into the corresponding register and processed as directed by the instruction. Since the input signal lines are at a "float" state, the logic level on them will be read as either a 1 or a 0 for that pin position. A low (≥0.8V) input causes a logic 0 to be read and a high (≥2.0V) input causes a logic 1 to be read. The output values can also be read (if the direction bit = 1) since the outputs are also on the pins. The input receivers are TTL compatible, are not latched, and are sampled near the end of each clock cycle ∅2 and gated onto the internal bus when selected.

PORT A NIBBLE ADDRESSING

Whenever a port is shared as an output, care must be exercised that one CPU does not destroy the other CPU's output data. In general, this can be avoided by allocating complete output ports to each CPU so that there is no possibility of conflict. However, there may be some situations where at least one port must be shared for outputs to get the proper mix for the required application. Port A is slightly different from the other bidirectional ports to allow port A to be safely shared as an output port by both CPU's.

Port A is divided into two 4-bit "nibble ports". Each half (nibble) of Port A may be independently addressed by each CPU as defined by two bits in the Bus Control Register (BCR2 and BCR3) as described in Bus Extension and Host Interface section. Depending upon the control bits, either CPU may be assigned to write to both halves, write to neither half (only read—the other CPU writes to the whole register), write to top half, or write to bottom half. When a mode has been selected for writing to only one-half of the port, the other half is unaffected.

ALTERNATIVE MODES OF OPERATION

Bidirectional Ports C, D, and F all have alternative modes of operation which may be selected in lieu of the bidirectional port capabilities.

Port C is a data bus for a host computer when the R65C00/21 is being used as a programmable peripheral device. This is discussed in more detail under Bus Extension and Host Interface.

Port D is a multiplexed data bus (D0 through D7) and address bus (A0 through A7) when the R65C00/21 is used as a microcomputer with external memory and I/O devices. This is also detailed under Bus Extension and Host Interface.

Port F also has the capability of operating in conjunction with other segments of the R65C00/21 architecture as described below.

PORT F CONTROL AND STATUS

The Interrupt Control and Status Register (ICSR) and the Clear Interrupt Flags Register (CIFR) control and monitor the operation of the Port F external interrupts (bits 2, 1, 0) as well as interprocessor communication interrupts.

When the PF0 edge-sensitive circuit detects a positive transition, bit 4 of the ICSR is set to a 1. An internal interrupt request ($\overline{IRQ}$) is generated to a CPU whenever this bit is set and the corresponding PF0 Interrupt Enable Flag (ICSR bit 0) is set to a 1 for that CPU. Similarly, a negative going transition on PF1 sets the edge detect flag in ICR bit 5. ICSR bit 1 is the corresponding PF1 Interrupt Enable bit. As in all cases of the interrupt enable bits, each CPU has its own set, addressed at the same location, but held separately.

Port F signal PF2 has an external interrupt request ($\overline{IRQ}$) capability. When this signal goes low, bit 6 of the Interrupt Control and Status Register is set and remains set as long as the signal is low. If the corresponding PF2 Interrupt Enable bit (bit 2) in its segment of the Interrupt Control and Status Register is a 1 while the PF2 Low Interrupt bit (bit 6) is a 1, an interrupt request is generated.

Each CPU may thus control the external interrupt independently of the internal interrupts. If the I flag in the Processor Status Register of a particular CPU is a 1, no $\overline{IRQ}$'s will be honored. If the I flag is a 0 and that CPU's interrupt enable in bit 2 of the Interrupt Control and Status Register is a 0, only internal interrupts will interrupt that CPU. If bit 2 is a 1, any $\overline{IRQ}$ will be honored.

The Port F signals PF3 and PF4 can be used as external interfaces for Counter/Timers A and B, respectively (refer to the Counter/Timers description). Finally, PF7 can be used as an active-low interrupt to a host processor. The operation of the R65C00/21 with a host processor is discussed under Bus Extension mode.

The Inter-Processor Communication Interrupt (IPCA and IPCB) bit in the ICSR allows each CPU to interrupt the other CPU if all of the other normal IRQ conditions are correct. CPU A sets the IPCB Interrupt Flag in CPU B's Interrupt Control and Status Register and CPU B sets the IPCA Interrupt Flag in CPU A by any write to location 0014, the Inter-Processor communications Interrupt Register. This is not an actual register, but writing any value here sets the other CPU IPCI flag. This inter-processor communications is illustrated in Figure 7.

Interrupt Control And Status Register (ICSR)

7	6	5	4	3	2	1	0
IPCA INT FLAG	PF2 LOW INT FLAG	PF1 NEG EDGE INT FLAG	PF0 POS EDGE INT FLAG	IPCA INT ENBL	PF2A INT ENBL	PF1A INT ENBL	PF0A INT ENBL
IPCB INT FLAG				IPCB INT ENBL	PF2B INT ENBL	PF1B INT ENBL	PF0B INT ENBL

Bit 7 **Inter-Processor Communication (IPC) Interrupt Flag (A or B)**
1 An inter-processor interrupt is requested by the other CPU
0 No internal interrupt is requested

Bit 6 **PF2 Low Interrupt Flag (A and B)**
1 PF2 is low
0 PF2 is high

Bit 5 **PF1 Negative Edge Detect Interrupt Flag**
1 A positive-to-negative transition on PF1 occurred
0 No positive-to-negative transition on PF1 occurred

Bit 4 **PF0 Positive Edge Detect Interrupt Flag**
1 A positive-to-negative transition on PF0 occurred
0 No positive-to-negative transition on PF0 occurred

Bit 3 **Inter-Processor Communication Interrupt Enable (A or B)**
1 Enables inter-processor communication interrupt (bit 7)
0 Disables inter-processor communication interrupt (bit 7)

Bit 2 **PF2 Interrupt Enable (A or B)**
1 Enables PF2 interrupt (bit 6)
0 Disables PF2 interrupt (bit 6)

Bit 1 **PF1 Interrupt Enable (A or B)**
1 Enables PF1 interrupt (bit 5)
0 Disables PF1 interrupt (bit 5)

Bit 0 **PF0 Interrupt Enable (A or B)**
1 Enables PF0 interrupt (bit 4)
0 Disables PF0 interrupt (bit 4)

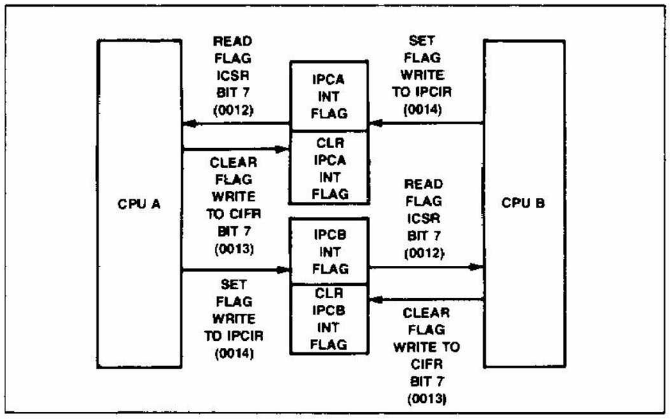

CPU A
READ FLAG ICSR BIT 7 (0012)
IPCA INT FLAG
SET FLAG WRITE TO IPCIR (0014)
CLR IPCA INT FLAG
CLEAR FLAG WRITE TO CIFR BIT 7 (0013)
READ FLAG ICSR BIT 7 (0012)
CPU B
IPCB INT FLAG
SET FLAG WRITE TO IPCIR (0014)
CLR IPCB INT FLAG
CLEAR FLAG WRITE TO CIFR BIT 7 (0013)

Figure 7. Inter Processor Communication

CLEAR INTERRUPT FLAGS REGISTER (CIFR)

The Clear Interrupt Flags Register (CIFR) is similar to the ICSR in that only one address is used but the bit pattern operates only on the status bits for its own processor. Thus only CPU A may clear IPCA but either may clear the edge detection flag bits. Bit 6 will only be cleared when the signal on PF2 goes high. Actually, the Clear Interrupt Flags Register is not a register at all, but addressing a bit pattern to this location performs the function. Any bit to which a zero is written will clear the corresponding interrupt flag. A read of this word returns logic one's so that the new Reset Memory Bit instructions may be used to clear these flags.

Clear Interrupt Flags Register (CIFR)

7	6	5	4	3	2	1	0
CLR IPCA INT FLAG	NOT USED	CLR PF1 NEG INT FLAG	CLR PF0 POS INT FLAG	NOT USED			
CLR IPCB INT FLAG							

Bit 7 **Clear Inter-Processing Communication Interrupt Flag**
1 Has no effect on the IPC Flag
0 Clears the IPC Interrupt Flag (specific CPU, A or B)

Bit 6 **Not Used**

Bit 5 **Clear PF1 Interrupt Flag**
1 Has no effect on the PF1 Interrupt Flag
0 Clears the PF1 Interrupt Flag (either CPU)

Bit 4 **Clear PF0 Interrupt Flag**
1 Has no effect on the PFO Interrupt Flag
0 Clears the PF0 Interrupt Flag (either CPU)

Bit 3-0 **Not Used**

OUTPUT ONLY PORT E

The output characteristics of Port E are identical to that of the bidirectional ports. The main difference is that there is no data direction register and also no capability of reading the information being output. Attempting to read Port E loads indeterminate data onto the internal bus.

Port E is a dual function port which, in addition to being an output port, can also serve as address bits A15 through A8 when the R65C00/21 is addressing external memory and I/O devices. This is discussed in more detail under Bus Extension and Host Interface.

INPUT ONLY PORT G

The input characteristics of the 4-bit Port G are the same as a bidirectional port in an input mode. The difference is that only four bits are input into the least significant bits of the data register and the most significant bits are loaded as zeros.

COUNTER/TIMERS

There are two separate 16-bit counter/timer systems in the R65C00/21: Counter/Timer A and Counter/Timer B. The block diagram of the counter/timers (also referred to as the timers, the counters, Timer A, or Timer B) is shown in Figure 8. Timer A has eight operating modes and five registers while Timer B has four operating modes and four registers. Both counter/timers have a 16-bit counter comprised of two 8-bit segments: Lower Counter (LCA and LCB, where A and B refer to Counter/Timer A and B) and Upper Counter (UCA and UCB). Both counter/timers also have a 16-bit latch section consisting of two 8-bit segments: Lower Latch (LLA and LLB) and Upper Latch (ULA and ULB). In addition, only Timer A has an 8-bit Snapshot Latch (SLA) register.

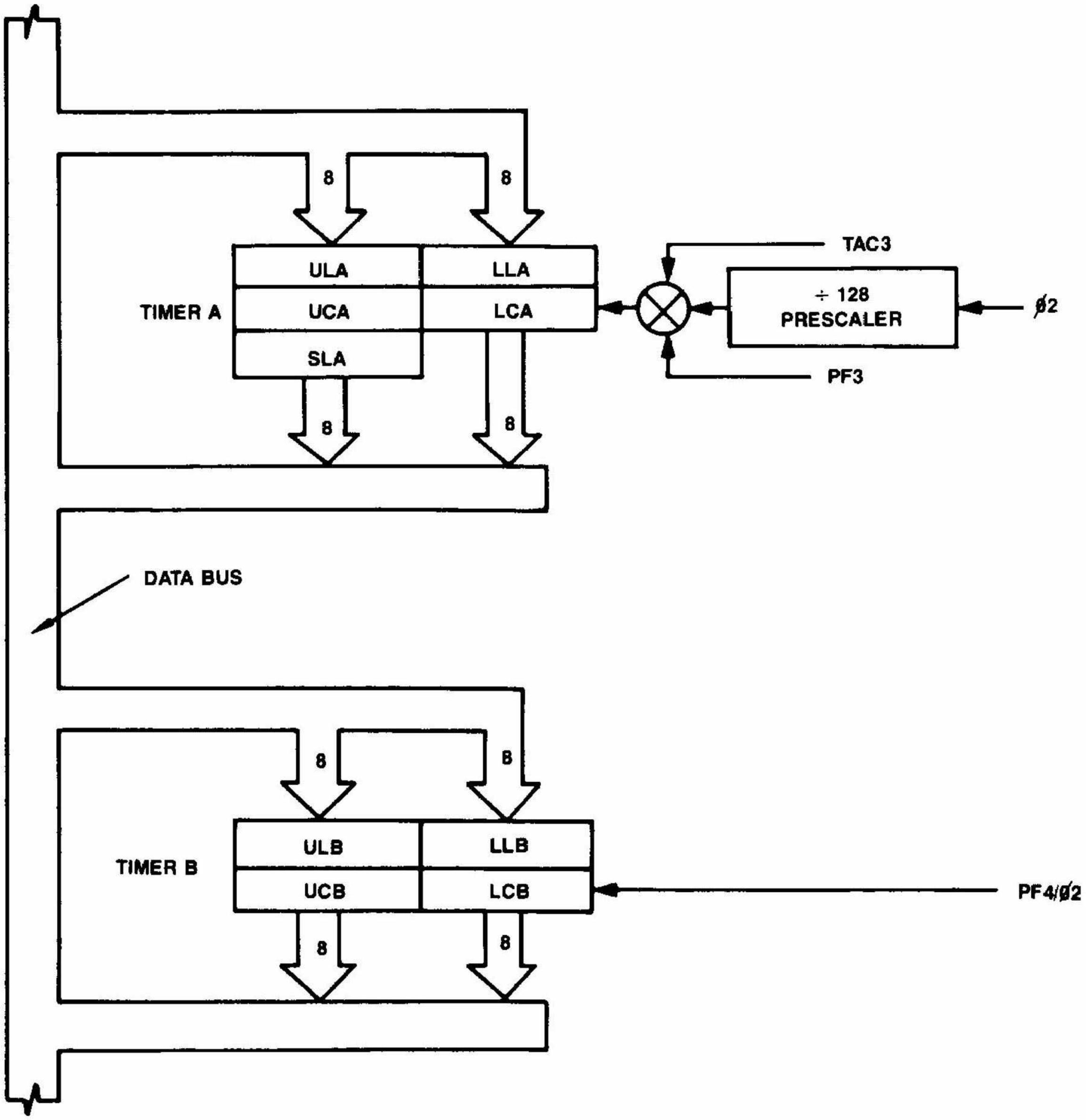


Figure 8. Counter/Timer Block Diagram

Data are written to the latches which act as holding registers for loading or reloading the initial counter/timer values upon mode initiation or counter/timer restart.

Both timers count down from the pre-set latch value and set an appropriate underflow flag when the counter counts through zero. The counter actually never counts below zero. At the time the counter would go negative, the contents of the latches replace the count value with no time delay.

Each counter/timer has three addresses for accessing the five (Counter/Timer A) or four (Counter/Timer B) 8-bit registers in its system. Consequently, the R/$\overline{W}$ line also aids in addressing the registers. Reading or writing to specific registers may also have other effects such as clearing an interrupt flag or transferring latch data to the counter. Consult the input/output and control register memory map in Table 2 for the effects of reading or writing to specific registers in the two counter/timer systems.

Each counter/timer has operating modes which are clocked either at the system clock rate (Ø2) or an external event clock rate. In addition, Timer A can operate with a prescaled Ø2/128 clock rate.

COUNTER/TIMER A (TA)

Counter/Timer A, with its four additional modes and Snapshot Latch, is generally more flexible than Counter/Timer B.

The Snapshot Latch (SLA) solves a problem which sometimes occurs when a timer is read. The problem is that between the time when the low byte of the 16-bit counter is read and the time when the high byte is read it is possible for the high byte to have been decremented. The resulting 16-bit value would, in this case be incorrect. In many modes of timer, the values are not actually read but the zero count transition is important. These types of applications do not require the use of the Snapshot Latch register. If the timer count value is to be used directly from a running timer, however, the Timer A Snapshot Latch should be used.

Timer A overcomes the problem stated above by sampling the value of the upper counter byte into the Snapshot Latch every time the lower counter byte is read. The value of the Upper Counter can be obtained by first reading the Lower Counter at address 0017, then reading the Snapshot Latch at address 0018 or 0019. Note that reading address 0019 also resets the Timer A Underflow (UFA) flag.

A second architectural difference between the two timers is that Timer A can have its clock input scaled down by a factor of 128 during normal power operation. This allows Timer A to measure longer periods of time internally while the microcomputer in operating at the Ø2 system clock rate. With a 4 MHz system clock, more than two second time intervals (up to 2.097 seconds) can be measured directly without any software intervention. Without the prescaler, 16.384 ms is the longest time interval at 4 MHz.

Timer A Mode Control

The operation of Timer A is controlled and monitored by the Timer A Control and Status Register (TACRS).

Bits 0-2 select the Timer A mode of operation.

Bit 3, when set to a 1, causes the clock prescaler to be switched into the circuit so that the timer may count longer intervals in modes which allow it.

Timer A Interrupt Enable, TACSR bit 4, if set to a 1 by a CPU, enables generation of an internal $\overline{IRQ}$ to that CPU when the UFA flag is set.

Bit 6 copies bit 3 of Port F (PF3).

Bit 7 is the UFA bit which indicates that Timer A has counted down through zero. This may be detected by reading the bit or may be used to cause an $\overline{IRQ}$ interrupt if bit 4 of the TACSR is set to a 1. The UFA flag is reset to a 0 by reading SLA or writing ULA at address 0019.

Timer A Control And Status Register (TACSR)

7	6	5	4	3	2	1	0
TMR A UNFL FLAG (UFA)	PF3 LEVEL IND	NOT USED	TMR A INT ENBL	TMR A CLK PRESC SEL	TIMER A MODE SELECT		

Bit 7	**Timer A Underflow Flag (UFA)**
1	Underflow condition occurred
0	No underflow
Bit 6	**Port F Bit 3 (PF3) Level**
1	PF3 High
0	PF3 Low
Bit 5	**Not Used** (Don't care)
Bit 4	**Timer A Interrupt Enable**
1	Enable Timer A Interrupt
0	Disable Timer A Interrupt
Bit 3	**Timer A Clock Prescaler Enable**[1]
1	Enable Clock Prescaler (Ø2/128)
0	Disable Clock Prescaler (Ø2)

Bits 2 to 0 **Timer A Mode Select (TAMS)**

2	1	0	
0	0	0	Timer A Off
0	0	1	Free-Run Event Counter Mode[1]
0	1	0	Free-Run Pulse Width Measurement Mode[1]
0	1	1	Retriggerable One-Shot Timer Mode[1]
1	0	0	One-Shot Interval Timer Mode
1	0	1	Free-Run Interval Timer Mode
1	1	0	One-Shot Pulse Generation Mode
1	1	1	Free-Run Pulse Generation Mode

Note:

1. Prescaler must be disabled (bit 3=0) for Free-Run Event Counter Mode, Free-Run Pulse Width Measurement Mode, and Retriggerable One-Shot Timer Mode. These three modes do not allow prescaling.

Timer A Operating Modes

The Timer A mode of operation is selected by setting bits 0-2 of the Timer A Control and Status Register (TACSR) to the appropriate code.

Timer A Off, Mode 0

Timer A is turned off in this mode. The Timer A Underflow Flag (UFA) stays at its current state. The counter holds its current value and may be read. Writing to the registers performs the usual functions associated with that address but the counter remains stopped. This is the default condition.

Timer A Free-Run Event Counter, Mode 1

The Timer A Upper Counter (UCA) and Lower Counter (LCA) is loaded with the Timer A Upper Latch (ULA) and Lower Latch (LLA) value when the data is written to the Timer A Upper Latch at address 0019. Timer A then decrements by 1 at each negative transition of the signal on input Port PF3. (The Port F data direction register must have a 0 in bit 3.) The Timer A Underflow Flag (UFA) is set to 1 when the counter decrements below zero. At this same time, the latch value is reloaded into UCA and LCA. The maximum rate of the signal on PF3 which may be detected is one-half of the Ø2 system clock rate.

Timer A Free-Run Pulse Width Measurement, Mode 2

Writing to ULA at 0019 transfers the 16-bit latch to the counter which operates as a timer in this mode. The initial value in the timer is decremented at the Ø2 rate when the PF3 signal is low. Otherwise, the counter holds its value. Counting stops when the PF3 signal goes high and will resume if the signal goes low again. If the counter counts below zero, the counter initial value is reloaded from the latches and the UFA flag is set.

Timer A One-Shot Retriggerable Timer, Mode 3

This mode is similar to Mode 4 except that the timer restarts each time PF3 goes through a high-to-low transition and counts down until the counter goes through zero. A second difference is that the clock prescaler may not be used with this mode. The data direction register bit 3 (PF3) must be zero to select input.

Timer A One-Shot Interval Timer, Mode 4

Writing to ULA at 0019 transfers the initial value from the latches and starts the timer. The timer counts at either the Ø2, or scaled Ø2 (Ø2/128), rate. When the counter counts through zero, the latch value is transferred to the counter, the UFA flag is set and the counter stops counting.

Timer A Free-Run Interval Timer, Mode 5

Writing ULA at 0019 transfers the 16-bit latch value to the timer and starts it running. The counter counts down at either the Ø2, or the scaled Ø2 (Ø2/128), rate. When the counter counts through zero the UFA flag is set, the value in the latches is transferred to the counter, and the counter continues to count down.

Timer A One-Shot Pulse Generation, Mode 6

The PF3 data direction register bit must be set to a 1 before starting this mode to initially force a high output. Writing ULA at 0019 starts the timer and clears the PF3 data output bit to a 0 causing a low output. The PF3 output remains low until the timer counts through zero. At this time, the PF3 output goes high until the mode is restarted or a new mode is selected. The UFA flag is also set at this time and the counter is stopped. The timer counts at either the Ø2, or the scaled Ø2 (Ø2/128), rate.

Timer A Free-Run Pulse Generation, Mode 7

The data direction register for PF3 must be set to a 1 to select the PF3 output before starting this mode. Writing to ULA at 0019 sets PF3 to 0 forcing a low output and starts the timer. Each time the timer counts through zero, the PF3 output changes state to generate a square wave at a rate dependent upon the latch value. The timer counts at either Ø2, or the scaled Ø2 (Ø2/128), rate. Each time the counter counts through zero, the latch contents are automatically transferred to the timer registers and the UFA flag is set.

COUNTER/TIMER B (TB)

Timer B is a simpler timer than Timer A but it still retains great flexibility. Unlike Timer A, there is no "off" mode (the default mode is the Free-Run Interval Timer Mode) and there is no separate selectable clock prescaler. All counting (except for counting external events) is done either at the Ø2 clock rate or Ø2/128 rate (when low power mode is selected). Another difference is that Timer B does not have the snapshot latch register for freezing the upper timer byte for reading. However, in its normal modes the counter counts through zero to set the Underflow Flag B (UFB) so that a snapshot latch register is not required.

Timer B Mode Control

The operation of Timer B is controlled and monitored by the Timer B Control and Status Register (TBCSR).

Bits 0-1 select the Timer B operating mode.

Timer B Interrupt Enable, bit 4, when set to a 1 by a CPU, enables generation of an internal interrupt request ($\overline{\text{IRQ}}$) to that CPU when the UFB flag is set.

Bit 6 of the TBCSR copies bit 4 of Port F (PF4).

Bit 7 in the TBCSR is the UFB bit which indicates that Timer B has counted down through zero. This may be detected by reading the bit or may be used to cause an $\overline{\text{IRQ}}$ interrupt if bit 4 of the TBCSR is set to a 1. The UFB bit is reset by either reading UCB or writing to ULB at address 001D.

Timer B Control and Status Register (TBCSR)

7	6	5	4	3	2	1	0
TMR B UNFL FLAG (UFB)	PF4 LEVEL IND	NOT USED	TMR B INT ENBL	NOT USED		TIMER B MODE SELECT	

Bit 7	**Timer B Underflow Flag (UFB)**
1	Underflow condition occurred
0	No underflow
Bit 6	**Port F Bit 4 (PF4) Level Indicator**
1	PF4 High
0	PF4 Low
Bit 5	**Not Used** (Don't care)
Bit 4	**Timer B Interrupt Enable**
1	Enable Timer B Interrupt
0	Disable Timer B Interrupt
Bits 3-2	**Not Used** (Don't care)
Bits 1-0	**Timer B Mode Select (TMS)**
1 0	
0 0	Free-Run Interval Timer Mode
0 1	Free-Run Pulse Generator Mode
1 0	Event Counter Mode
1 1	Pulse Width Measurement Mode

Timer B Operating Modes

The Timer B operating mode is selected by setting bits 0 and 1 in the TBCSR to the appropriate code.

Timer B Free-Run Interval Timer, Mode 0

Writing to Timer B Upper Latch (ULB) at 001D transfers the 16-bit latch value to the timer and starts it running. The counter counts down at the Ø2 rate. When the counter counts through zero, the Timer B Underflow Flag (UFB) is set to a 1, the value in the latches is transferred to the counter and the counter continues to count down.

Timer B Free-Run Pulse Generation, Mode 1

The data direction register for PF4 must be set to a 1 to select PF4 output before starting this mode. Writing to ULB at 001D sets PF4 to 0 to force the PF4 output low and starts the timer. Each time the timer counts through zero, the PF4 output changes state to generate a square wave at a rate dependent upon the initial value loaded into the latches. The timer counts at the Ø2 rate. Each time the counter counts through zero, the latch values are automatically transferred to the timer registers and the UFB flag is set to a 1.

Timer B Event Counter, Mode 2

The data direction register bit for PF4 must be set to a 0 to select PF4 input prior to selecting this mode. The counter is loaded with the latch value when the ULB data is written to address 001D. Timer B then decrements by 1 at each negative transition on input Port PF4. The Timer B Underflow Flag (UFB) is set to a 1, when Counter B counts through zero. At this same time, the latch value is reloaded into Timer B. The maximum rate of the signal on PF4 which may be detected is one-half of the Ø2 clock rate.

Timer B Pulse Width Measurement, Mode 3

Writing to ULB at 001D transfers the 16-bit latch value to the counter. The initial value in the timer is decremented at the Ø2 rate when the PF4 signal is low. Each time the PF4 signal goes high, the counter stops and then continues when the signal is low again. If the counter counts through zero, the UFB flag is set to 1 and the latch value transfers to reinitialize the counter and the countdown continues as long as PF4 is low.

BUS EXTENSION

In addition to its application as a single-chip microcomputer, the bus extension mode allows the R65C00/21 to operate as a microprocessor with external memory and I/O.

BUS EXTENSION MODE

When the R65C00/21 is used as a single-chip microcomputer, all of the output ports may be used as input or output ports. However, to use the R65C00/21 with external ROM, RAM, or I/O, a number of the ports act as extensions of the internal address and data buses. Specifically, Port D becomes dedicated as a multiplexed 8-bit data and address bus. Port D provides both the data bus (D0 through D7) and the low bits of the address (A0 through A7) on pins PD0 through PD7. When a bus extension mode is selected, the Port D Data Direction Register must be cleared to zero (its default condition) to configure Port D as all inputs. The R65C00/21 then controls Port D as an extension of the internal bus structure and provides an active-low External Memory Select ($\overline{EMS}$) strobe signal at the time the address bits are available. The $\overline{EMS}$ signal is present even when Port D is being used as a normal input/output register.

The R65C00/21 has the option of using 8-, 12- or 16-bit address bus extensions. Selection of the bus extension mode is controlled by bits 0 and 1 of the Bus Control Register (BCR). When the 8-bit mode is selected, only the Port D multiplexed address/data bus function is required. However, if either the 12- or 16-bit address bus extension is selected, either one half or all of output Port E also becomes dedicated to the bus extension function. If a 16-bit bus extension is selected, then all of Port E becomes the upper address bits A8 through A15 on pins PE0 through PE7, respectively. If the 12-bit bus extension is selected, then the address lines A8 through A11 appear on PE0 through PE3. In this case, PE4 through PE7 have their usual output function.

Since Port D is multiplexed, it is necessary that external latches be supplied to hold the lower eight bits of the address bus. The EMS output is low when the address is being supplied from Port D. All of the other necessary control bus signals are also provided; these include Ø2 and R/$\overline{W}$. The SYNC and ØA signals are also brought out for use by development systems and bus analyzers for system debugging.

In a one-chip configuration, the 128 bytes of internal page one RAM (address 0180 through 01FF), is logically combined with page (0080-00FF). However, when an extended bus is used, the stack page may be addressed in its normal range in external memory (0100-01FF). When bit 4 of the Bus Control Register is a 0, page one is internal and shared with page zero; when it is a 1, page one is external allowing full 256 bytes available to the two stacks.

Figure 9 is an overall block diagram of a system using the R65C00/21 in the bus extension mode.

The CPU A Active signal (bit 7 of the BCR) is high when CPU A is controlling the system bus, and low when CPU B is active. This bit copies the state of the ØA output signal. Consequently, the bit may be sampled in common subroutines to determine the calling CPU, or for bank selection purposes. Thus, CPU A and CPU B may have some external memory or I/O dedicated to their exclusive use. Each may separately address as much as 59.5K bytes of external memory map, or external memory may be shared.

Bus Control Register (BCR)

7	6	5	4	3	2	1	0
CPU A ACTIVE	NOT USED		PAGE ONE EXT	PORT A NIBBLE MODE		BUS EXTENSION MODE	

Bit 7	CPU A Active
1	CPU A active
0	CPU B active

Bits 6-5	Not Used (Don't Care)

Bit 4	Page One External/Internal Mapping
1	Page One External
0	Page One Internal

Bits 3-2		Port A Write Nibble Control*
3	2	
0	0	CPU A writes to both halves (PAO-PA7).
0	1	CPU A writes to upper half (PA4-PA7); CPU B writes to lower half (PAO-PA3).
1	0	CPU A writes to lower half (PAO-PA3); CPU B writes to upper half (PA04-PA7).
1	1	CPU B writes to both halves (PAO-PA7).

Bits 1-0		Bus Extension Mode
1	0	
0	0	Bus Extension Mode not selected.
0	1	8-bit Address Extension Mode. Range equals 256.
1	0	12-bit Address Extension Mode. Range equals 4096.
1	1	16-bit Address Extension Mode. Range equals 65,536.

Note:
*Either CPU may read the full port at any time.

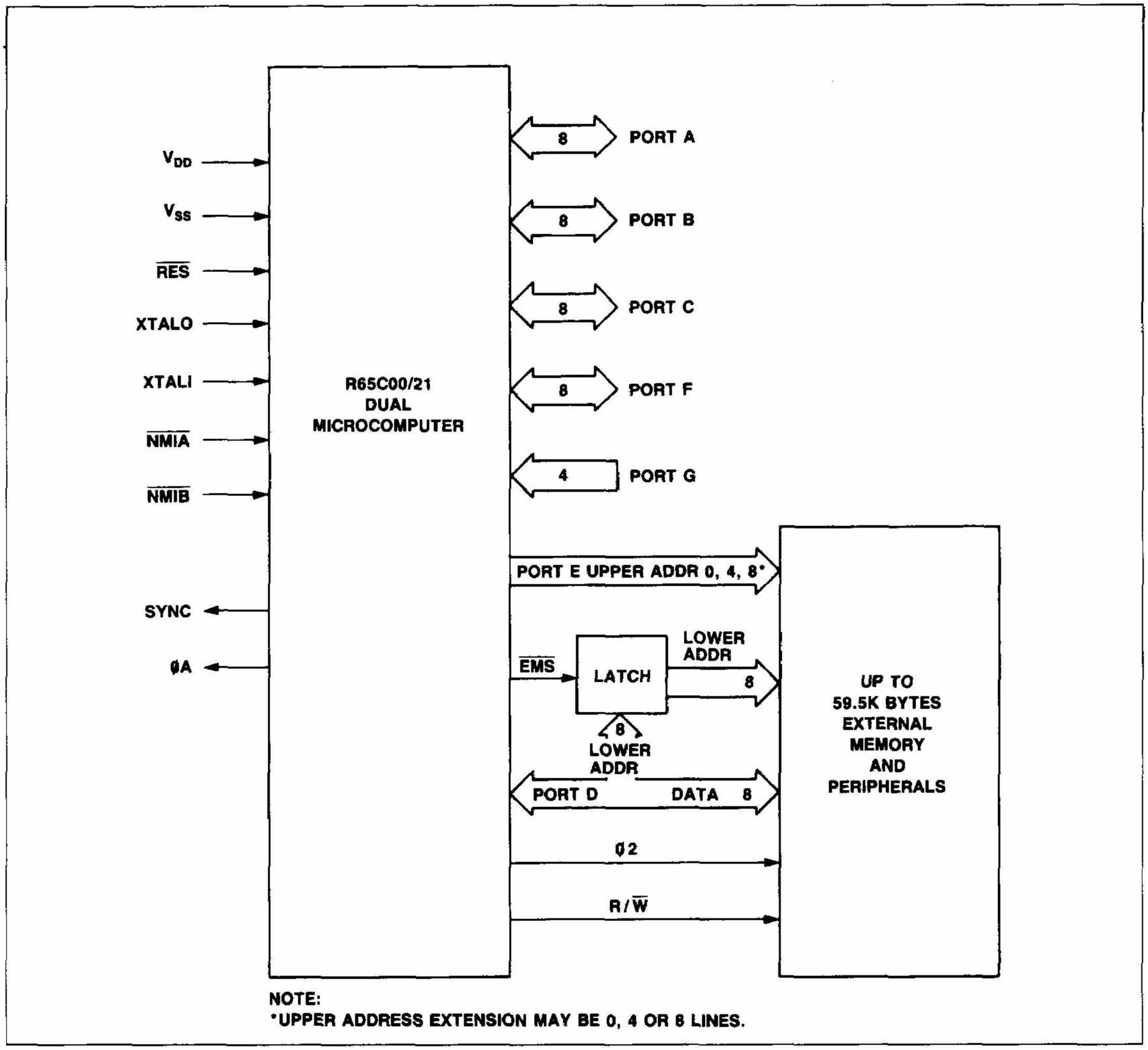


Figure 9. Bus Extension Mode Block Diagram

PROGRAMMABLE PERIPHERAL TO A HOST MODE

An overall block diagram of a system using an R65C00/21 as an intelligent controller is shown in Figure 10.

In this configuration, three of the R65C00/21 input/output ports have special significance. Port C becomes the interface with the host data bus (Port C's Data Direction Register must specify as the input; i.e., all zeros). Pin PF7 becomes an active-low Host Interrupt ($\overline{\text{HINT}}$) line, and the 4-bit input Port G becomes the control pins interface to the Host computer.

The R65C00/21 is configured to operate as a peripheral for either the R6500 or 6800 families, or the Z80 or 8080 families. When operating in the 6500/6800 mode, PG0 is an input for the host Ø2 (HØ2) and PG1 is the input for the host R/$\overline{\text{W}}$ (HRW) control lines.

When operating in the Z80/8080 mode, PG0 accepts the host $\overline{\text{RD}}$ ($\overline{\text{HRD}}$) control and PG1 provides the host $\overline{\text{WR}}$ ($\overline{\text{HWR}}$) control.

In both cases, PG2 serves as a register select (HRS) and PG3 acts as an active-low chip select ($\overline{\text{CS}}$) from the host. HRS is used in conjunction with the $\overline{\text{CS}}$ and $\overline{\text{HWR}}$ to control reading or writing of data or status information as shown in Table 4.

Control of the host mode options is provided by the Host Control and Status Register (HCSR).

When the host writes a byte into the Input Buffer (Port C), the Input Buffer Full (IBF) flag is set to a 1. Similarly, when a byte is read from the Output Buffer (Port C) by the host, the Output Buffer Full (OBF) flag is cleared to a 0. Setting bit 3 of the HCSR enables generation of an internal interrupt request (IRQ) when either the IBF flag is a 1 or the OBF flag is a 0. This logic is duplicated for both CPU's.

Setting bit 2 of the HCSR to a 1 enables generation of any interrupt signal to the host computer. In this case, bit 7 of Port F is pulled low by either a write to Port C (Output Buffer) or a read from Port C (Input Buffer), by either of the R65C00/21 CPU's.

Bit 5 of the HCSR is actually two different bits representing Register Select Input (RSI) and Register Select Output (RSO). The R65C00/21 writes bit RSO and reads bit RSI, while the host writes RSI and reads RSO. The R65C00/21 writes a 0 to this bit when Port C is addressed at 0002 and a 1 when Port C is addressed at 0003. When the host writes to the R65C00/21 through Port C, the level of the HRS input is copied into the RSI bit. This bit allows the communications between the host system and the R65C00/21 to flag the type of data being transferred so that command information may be distinguished from data.

Table 4. Register Select Control

$\overline{\text{CS}}$ (PG3)	HRS (PG2)	HRW (PG1)	HØ2 (PG0)	Host Function (6500/6800 Mode)
H	—	—	—	Host Interface Deselected
L	L	L	H	Write Input Buffer, HCSR5 RSI cleared, set IBF
L	L	H	H	Read Output Buffer, Clear OBF
L	H	L	H	Write Input Buffer, HCSR5 RS1 set, set IBF
L	H	H	H	Read upper 3 bits of HCSR; OBF, IBF & RSO
$\overline{\text{CS}}$ (PG3)	**HRS (PG2)**	**$\overline{\text{HWR}}$ (PG1)**	**$\overline{\text{HRD}}$ (PG0)**	**Host Function (8080/Z80 Mode)**
H	—	—	—	Deselected
L	L	L	H	Write Input Buffer, HCSR5 RSI cleared, set IBF
L	L	H	L	Read Output Buffer, Clear OBF
L	H	L	H	Write Input Buffer, HCSR5 RS1 set, set IBF
L	H	H	L	Read upper 3 bits of HCSR; OBF, IBF & RSO

Host Control and Status Register (HCSR)

7	6	5	4	3	2	1	0
O/P BUFF FULL INT FLAG (OBE)	I/O BUFF FULL INT FLAG (IBF)	I/O REG SEL (RSI) (RSO)	NOT USED	I/OA INT ENBL I/OB INT ENBL	HOST INT ENBL	HOST BUS ENBL	HOST BUS TYPE

Bit 7	**Output Buffer Empty (OBE) Flag**
1	Output Buffer Full
0	Output Buffer Empty
Bit 6	**Input Buffer Full (IBF) Flag**
1	Input Buffer Full
0	Input Buffer Empty
Bit 5	**Register Select**
—	Distinguishes commands from data. Host reads RSO and R65C00/21 reads RSI. Selection of 1 or 0 to represent commands or data is user defined.
Bit 4	**Not Used.** (Don't care)
Bit 3	**Input/Output Buffer Interrupt Enable**
1	Enable IRQ IBF = 1)
0	Disable IRQ
Bit 2	**Host Interrupt ($\overline{\text{HINT}}$) Output Enable**
1	Disable $\overline{\text{HINT}}$ Output to Host
0	Enable $\overline{\text{HINT}}$ Output to Host (OBF = 1)
Bit 1	**Host Bus Enable**
1	Disable Host Bus
0	Enable Host Bus
Bit 0	**Host Bus Type**
1	Host Bus is Z80/8080
0	Host Bus is 6500/6800

Note:
Register is cleared to all zeros by $\overline{\text{RES}}$.

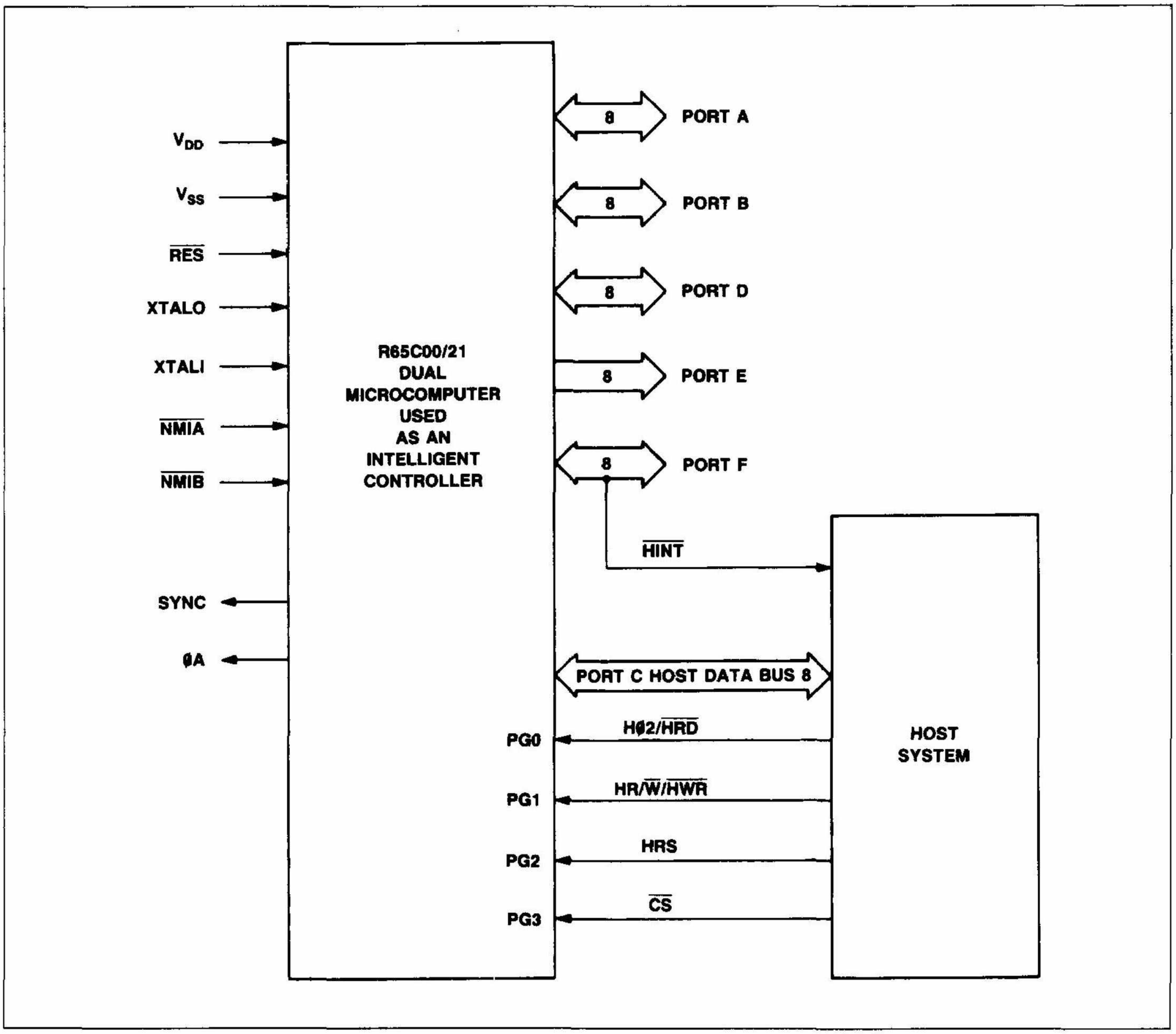


Figure 10. Host Mode Block Diagam

EMULATION MODE

The R65C00/21 can operate in an emulation mode under external signal control.

Emulation mode deselects the internal ROM and enables the 16-bit Expanded Bus mode, independent of the bus mode programmed in the Bus Control Register. Since the Expanded Bus mode uses peripheral Ports D and E, provision is made for these to be emulated in external hardware. This is accomplished by forcing all memory references to Ports D and E to be External Bus cycles. Accesses to the Data Direction Register for Port D are also forced external.

To further aid program development in emulation mode, all bus cycles which perform a memory or I/O write operation, whether the true destination is internal or external, will assert the External Memory Strobe (EMS) signal. This allows a copy of internal register and memory values to be kept in external memory.

Emulation mode is selected by applying the Ø2 output clock signal to the RES input pin.

INSTRUCTION SET IN ALPHABETIC SEQUENCE

The following table contains a summary of the R65C00/21 and R65C29 CPU instruction set. For detailed information, consult the R6502 Microcomputer System Programming Manual, Order No. 202.

The instructions notated with a * are added instructions for the R65C00/21 and R65C29 which are not part of the standard 6502 instruction set.

Instruction Set In Alphabetic Sequence

Mnemonic	Description	Mnemonic	Description
ADC	Add Memory to Accumulator with Carry	LDA	Load Accumulator with Memory
AND	"AND" Memory with Accumulator	LDX	Load Index X with Memory
ASL	Shift Left One Bit (Memory or Accumulator)	LDY	Load Index Y with Memory
		LSR	Shift One Bit Right (Memory or Accumulator)
*BRA	Branch Always		
*BBR	Branch on Bit Reset Relative	*MUL	Multiply
*BBS	Branch on Bit Set Relative		
BCC	Branch on Carry Clear	NOP	No Operation
BCS	Branch on Carry Set		
BEQ	Branch on Result Zero	ORA	"OR" Memory with Accumulator
BIT	Test Bits in Memory with Accumulator		
BMI	Branch on Result Minus	PHA	Push Accumulator or Stack
BNE	Branch on Result not Zero	PHP	Push Processor Status on Stack
BPL	Branch on Result Plus	*PHX	Push Index X
BRK	Force Break	*PHY	Push Index Y
BVC	Branch on Overflow Clear	PLA	Pull Accumulator from Stack
BVS	Branch on Overflow Set	PLP	Pull Processor Status from Stack
		*PLX	Pull Index X
CLC	Clear Carry Flag	*PLY	Pull Index Y
CLD	Clear Decimal Mode		
CLI	Clear Interrupt Disable Bit	*RMB	Reset Memory Bit
CLV	Clear Overflow Flag	ROL	Rotate One Bit Left (Memory or Accumulator)
CMP	Compare Memory and Accumulator	ROR	Rotate One Bit Right (Memory or Accumulator)
CPX	Compare Memory and Index X	RTI	Return from Interrupt
CPY	Compare Memory and Index Y	RTS	Return from Subroutine
DEC	Decrement Memory by One	SBC	Subtract Memory from Accumulator with Borrow
DEX	Decrement Index X by One	SEC	Set Carry Flag
DEY	Decrement Index Y by One	SED	Set Decimal Mode
		SEI	Set Interrupt Disable Status
EOR	"Exclusive-Or" Memory with Accumulator	*SMB	Set Memory Bit
		STA	Store Accumulator in Memory
INC	Increment Memory by One	STX	Store Index X in Memory
INX	Increment Index X by One	STY	Store Index Y in Memory
INY	Increment Index Y by One		
		TAX	Transfer Accumulator to Index X
JMP	Jump to New Location	TAY	Transfer Accumulator to Index Y
JSR	Jump to New Location Saving Return Address	TSX	Transfer Stack Pointer to Index X
		TXA	Transfer Index X to Accumulator
		TXS	Transfer Index X to Stack Register
		TYA	Transfer Index Y to Accumulator

INSTRUCTION SET SUMMARY

MNEMONIC	OPERATION	ADDRESSING MODE																																													PROCESSOR STATUS CODES							
		IMMEDIATE			ABSOLUTE			ZERO PAGE			ACCUM.			IMPLIED			(IND, X)			(IND), Y			Z. PAGE, X			ABS, X			ABS, Y			RELATIVE			INDIRECT			Z. PAGE, Y			BIT ADDRESSING (OP BY BIT #)								7	6	5	4	3	2	1	ø
		OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	ø	1	2	3	4	5	6	7	N	V	·	B	D	I	Z	C
ADC	A + M + C → A (4) (1)	69	2	2	6D	4	3	65	3	2							61	6	2	71	5	2	75	4	2	7D	4	3	79	4	3																		N	V	·	·	·	·	Z	C
AND	A ∧ M → A (1)	29	2	2	2D	4	3	25	3	2							21	6	2	31	5	2	35	4	2	3D	4	3	39	4	3																		N	·	·	·	·	·	Z	·
ASL	C ← [7 ø] ← ø				øE	6	3	ø6	5	2	øA	2	1										16	6	2	1E	7	3																					N	·	·	·	·	·	Z	C
BBR[#(ø-7)]	Branch on M_b = 0 (5)																																								øF	1F	2F	3F	4F	5F	6F	7F	·	·	·	·	·	·	·	·
BBS[#(ø-7)]	Branch on M_b = 1 (5)																																								8F	9F	AF	BF	CF	DF	EF	FF	·	·	·	·	·	·	·	·
BCC	Branch on C = ø (2)																															9ø	2	2															·	·	·	·	·	·	·	·
BCS	Branch on C = 1 (2)																															Bø	2	2															·	·	·	·	·	·	·	·
BEQ	Branch on Z = 1 (2)																															Fø	2	2															·	·	·	·	·	·	·	·
BIT	A ∧ M				2C	4	3	24	3	2																																							M_7	M_6	·	·	·	·	Z	·
BMI	Branch on N = 1 (2)																															3ø	2	2															·	·	·	·	·	·	·	·
BNE	Branch on Z = ø (2)																															Dø	2	2															·	·	·	·	·	·	·	·
BPL	Branch on N = ø (2)																															1ø	2	2															·	·	·	·	·	·	·	·
BRA	Branch Always (1)																															80	3	2															·	·	·	·	·	·	·	·
BRK	Break													00	7	1																																	·	·	·	1	·	1	·	·
BVC	Branch on V = ø (2)																															5ø	2	2															·	·	·	·	·	·	·	·
BVS	Branch on V = 1 (2)																															7ø	2	2															·	·	·	·	·	·	·	·
CLC	ø → C													18	2	1																																	·	·	·	·	·	·	·	ø
CLD	ø → D													D8	2	1																																	·	·	·	·	ø	·	·	·
CLI	ø → I													58	2	1																																	·	·	·	·	·	ø	·	·
CLV	ø → V													B8	2	1																																	·	ø	·	·	·	·	·	·
CMP	A − M	C9	2	2	CD	4	3	C5	3	2							C1	6	2	D1	5	2	D5	4	2	DD	4	3	D9	4	3																		N	·	·	·	·	·	Z	C
CPX	X − M	Eø	2	2	EC	4	3	E4	3	2																																							N	·	·	·	·	·	Z	C
CPY	Y − M	Cø	2	2	CC	4	3	C4	3	2																																							N	·	·	·	·	·	Z	C
DEC	M − 1 → M				CE	6	3	C6	5	2													D6	6	2	DE	7	3																					N	·	·	·	·	·	Z	·
DEX	X − 1 → X													CA	2	1																																	N	·	·	·	·	·	Z	·
DEY	Y − 1 → Y													88	2	1																																	N	·	·	·	·	·	Z	·
EOR	A⊻M → A (1)	49	2	2	4D	4	3	45	3	2							41	6	2	51	5	2	55	4	2	5D	4	3	59	4	3																		N	·	·	·	·	·	Z	·
INC	M + 1 → M				EE	6	3	E6	5	2													F6	6	2	FE	7	3																					N	·	·	·	·	·	Z	·
INX	X + 1 → X													E8	2	1																																	N	·	·	·	·	·	Z	·
INY	Y + 1 → Y													C8	2	1																																	N	·	·	·	·	·	Z	·
JMP	Jump to New Loc				4C	3	3																												6C	6	3												·	·	·	·	·	·	·	·
JSR	Jump Sub				2ø	6	3																																										·	·	·	·	·	·	·	·
LDA	M → A (1)	A9	2	2	AD	4	3	A5	3	2							A1	6	2	B1	5	2	B5	4	2	BD	4	3	B9	4	3																		N	·	·	·	·	·	Z	·
LDX	M → X (1)	A2	2	2	AE	4	3	A6	3	2																			BE	4	3							B6	4	2									N	·	·	·	·	·	Z	·
LDY	M → Y (1)	Aø	2	2	AC	4	3	A4	3	2													B4	4	2	BC	4	3																					N	·	·	·	·	·	Z	·
LSR	ø → [7 ø] → C				4E	6	3	46	5	2	4A	2	1										56	6	2	5E	7	3																					0	·	·	·	·	·	Z	C
NOP	No Operation													EA	2	1																																	·	·	·	·	·	·	·	·
MUL	A * Y → A_{15-8} Y_{7-0}													02	10	1																																	·	·	·	·	·	·	·	·
ORA	AVM → A (1)	ø9	2	2	øD	4	3	ø5	3	2							ø1	6	2	11	5	2	15	4	2	1D	4	3	19	4	3																		N	·	·	·	·	·	Z	·
PHA	A → Ms S − 1 → S													48	3	1																																	·	·	·	·	·	·	·	·
PHP	P → Ms S − 1 → S													08	3	1																																	·	·	·	·	·	·	·	·
PHX	X → Ms S − 1 → S													DA	3	1																																	·	·	·	·	·	·	·	·
PHY	Y → Ms S − 1 → S													5A	3	1																																	·	·	·	·	·	·	·	·
PLA	S + 1 → S Ms → A													68	4	1																																	N	·	·	·	·	·	Z	·
PLP	S + 1 → S Ms → P													28	4	1																																	(Restored)							
PLX	S + 1 → S Ms → X													FA	4	1																																	N	·	·	·	·	·	Z	·
PLY	S + 1 → S Ms → Y													7A	4	1																																	N	·	·	·	·	·	Z	·
RMB[#(ø-7)]	ø → M_b (5)																																								ø7	17	27	37	47	57	67	77	·	·	·	·	·	·	·	·
ROL	← [7 ø] ← [C] ←				2E	6	3	26	5	2	2A	2	1										36	6	2	3E	7	3																					N	·	·	·	·	·	Z	C
ROR	→ [C] → [7 ø] →				6E	6	3	66	5	2	6A	2	1										76	6	2	7E	7	3																					N	·	·	·	·	·	Z	C
RTI	Rtrn Int													4ø	6	1																																	(Restored)							
RTS	Rtrn Sub													6ø	6	1																																	·	·	·	·	·	·	·	·
SBC	A − M − C̄ → A (1) (5)	E9	2	2	ED	4	3	E5	3	2							E1	6	2	F1	5	2	F5	4	2	FD	4	3	F9	4	3																		N	V	·	·	·	·	Z	C̄
SEC	1 → C													38	2	1																																	·	·	·	·	·	·	·	1
SED	1 → D													F8	2	1																																	·	·	·	·	1	·	·	·
SEI	1 → I													78	2	1																																	·	·	·	·	·	1	·	·
SMB[#(ø-7)]	1 → M_b (5)																																								87	97	A7	B7	C7	D7	E7	F7	·	·	·	·	·	·	·	·
STA	A → M				8D	4	3	85	3	2							81	6	2	91	6	2	95	4	2	9D	5	3	99	5	3																		·	·	·	·	·	·	·	·
STX	X → M				8E	4	3	86	3	2																												96	4	2									·	·	·	·	·	·	·	·
STY	Y → M				8C	4	3	84	3	2													94	4	2																								·	·	·	·	·	·	·	·
TAX	A → X													AA	2	1																																	N	·	·	·	·	·	Z	·
TAY	A → Y													A8	2	1																																	N	·	·	·	·	·	Z	·
TSX	S → X													BA	2	1																																	N	·	·	·	·	·	Z	·
TXA	X → A													8A	2	1																																	N	·	·	·	·	·	Z	·
TXS	X → S													9A	2	1																																	·	·	·	·	·	·	·	·
TYA	Y → A													98	2	1																																	N	·	·	·	·	·	Z	·

Notes:

1. Add 1 to N if page boundary is crossed
2. Add 1 to N if branch occurs to same page
   Add 2 to N if branch occurs to different page
3. Carry not (C̄) = Borrow
4. If in decimal mode Z flag is invalid
   accumulator must be checked on zero result.
5. Effects 8-bit data field of the specified zero page address.

LEGEND

X	−	Index X
Y	=	Index Y
A	=	Accumulator
M	=	Memory per effective address
M_s	=	Memory per stack pointer
M_b	=	Selecter zero page memory bit
M_7	=	Memory Bit 7
M_6	=	Memory Bit 6
+	=	Add
−	=	Subtract
∧	=	And
V	=	Or
⊻	=	Exclusive or
m	=	Number of cycles
#	=	Number of Bytes

INSTRUCTION SET OPERATION CODE MATRIX

The following matrix shows the op codes associated with the R65C00/21 and R65C29 CPUs. The matrix identifies the hexadecimal code, the mnemonic code, the addressing mode, the number of instruction bytes, and the number of machine cycles associated with each op code. Also, refer to the instruction set summary for additional information on these op codes.

MSD \ LSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	BRK Implied 1 7	ORA (IND, X) 2 6	MUL Implied 1 10			ORA ZP 2 3	ASL ZP 2 5	RMB0 ZP 2 5	PHP Implied 1 3	ORA IMM 2 2	ASL Accum 1 2			ORA ABS 3 4	ASL ABS 3 6	BBR0 ZP 3 5**	0
1	BPL Relative 2 2**	ORA (IND), Y 2 5*				ORA ZP, X 2 4	ASL ZP, X 2 6	RMB1 ZP 2 5	CLC Implied 1 2	ORA ABS, Y 3 4*				ORA ABS, X 3 4*	ASL ABS, X 3 7	BBR1 ZP 3 5**	1
2	JSR Absolute 3 6	AND (IND, X) 2 6			BIT ZP 2 3	AND ZP 2 3	ROL ZP 2 5	RMB2 ZP 2 5	PLP Implied 1 4	AND IMM 2 2	ROL Accum 1 2		BIT ABS 3 4	AND ABS 3 4	ROL ABS 3 6	BBR2 ZP 3 5**	2
3	BMI Relative 2 2**	AND (IND, Y) 2 5*				AND ZP, X 2 4	ROL ZP, X 2 6	RMB3 ZP 2 5	SEC Implied 1 2	AND ABS, Y 3 4*				AND ABS, X 3 4*	ROL ABS, X 3 7	BBR3 ZP 3 5**	3
4	RTI Implied 1 6	EOR (IND, X) 2 6				EOR ZP 2 3	LSR ZP 2 5	RMB4 ZP 2 5	PHA Implied 1 3	EOR IMM 2 2	LSR Accum 1 2		JMP ABS 3 3	EOR ABS 3 4	LSR ABS 3 6	BBR4 ZP 3 5**	4
5	BVC Relative 2 2**	EOR (IND), Y 2 5*				EOR ZP, X 2 4	LSR ZP, X 2 6	RMB5 ZP 2 5	CLI Implied 1 2	EOR ABS, Y 3 4*	PHY Implied 1 3			EOR ABS, X 3 4*	LSR ABS, X 3 7	BBR5 ZP 3 5**	5
6	RTS Implied 1 6	ADC (IND, X) 2 6				ADC ZP 2 3	ROR ZP 2 5	RMB6 ZP 2 5	PLA Implied 1 4	ADC IMM 2 2	ROR Accum 1 2		JMP Indirect 3 5	ADC ABS 3 4	ROR ABS 3 6	BBR6 ZP 3 5**	6
7	BVS Relative 2 2**	ADC (IND, Y) 2 5*				ADC ZP, X 2 4	ROR ZP, X 2 6	RMB7 ZP 2 5	SEI Implied 1 2	ADC ABS, Y 3 4*	PLY Implied 1 4			ADC ABS, X 3 4*	ROR ABS, X 3 7	BBR7 ZP 3 5**	7
8	BRA Relative 2 3*	STA (IND, X) 2 6			STY ZP 2 3	STA ZP 2 3	STX ZP 2 3	SMB0 ZP 2 5	DEY Implied 1 2		TXA Implied 1 2		STY ABS 3 4	STA ABS 3 4	STX ABS 3 4	BBS0 ZP 3 5**	8
9	BCC Relative 2 2**	STA (IND, Y) 2 6			STY ZP, X 2 4	STA ZP, X 2 4	STX ZP, Y 2 4	SMB1 ZP 2 5	TYA Implied 1 2	STA ABS, Y 3 5	TXS Implied 1 2			STA ABS, X 3 5		BBS1 ZP 3 5**	9
A	LDY IMM 2 2	LDA (IND, X) 2 6	LDX IMM 2 2		LDY ZP 2 3	LDA ZP 2 3	LDX ZP 2 3	SMB2 ZP 2 5	TAY Implied 1 2	LDA IMM 2 2	TAX Implied 1 2		LDY ABS 3 4	LDA ABS 3 4	LDX ABS 3 4	BBS2 ZP 3 5**	A
B	BCS Relative 2 2**	LDA (IND), Y 2 5*			LDY ZP, X 2 4	LDA ZP, X 2 4	LDX ZP, Y 2 4	SMB3 ZP 2 5	CLV Implied 1 2	LDA ABS, Y 3 4*	TSX Implied 1 2		LDY ABS, X 3 4*	LDA ABS, X 3 4*	LDX ABS, Y 3 4*	BBS3 ZP 3 5**	B
C	CPY IMM 2 2	CMP (IND, X) 2 6			CPY ZP 2 3	CMP ZP 2 3	DEC ZP 2 5	SMB4 ZP 2 5	INY Implied 1 2	CMP IMM 2 2	DEX Implied 1 2		CPY ABS 3 4	CMP ABS 3 4	DEC ABS 3 6	BBS4 ZP 3 5**	C
D	BNE Relative 2 2**	CMP (IND), Y 2 5*				CMP ZP, X 2 4	DEC ZP, X 2 6	SMB5 ZP 2 5	CLD Implied 1 2	CMP ABS, Y 3 4*	PHX Implied 1 3			CMP ABS, X 3 4*	DEC ABS, X 3 7	BBS5 ZP 3 5**	D
E	CPX IMM 2 2	SBC (IND; X) 2 6			CPX ZP 2 3	SBC ZP 2 3	INC ZP 2 5	SMB6 ZP 2 5	INX Implied 1 2	SBC IMM 2 2	NOP Implied 1 2		CPX ABS 3 4	SBC ABS 3 4	INC ABS 3 6	BBS6 ZP 3 5**	E
F	BEQ Relative 2 2**	SBC (IND), Y 2 5*				SBC ZP, X 2 4	INC ZP, X 2 6	SMB7 ZP 2 5	SED Implied 1 2	SBC ABS, Y 3 4*	PLX Implied 1 4			SBC ABS, X 3 4*	INC ABS, X 3 7	BBS7 ZP 3 5**	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

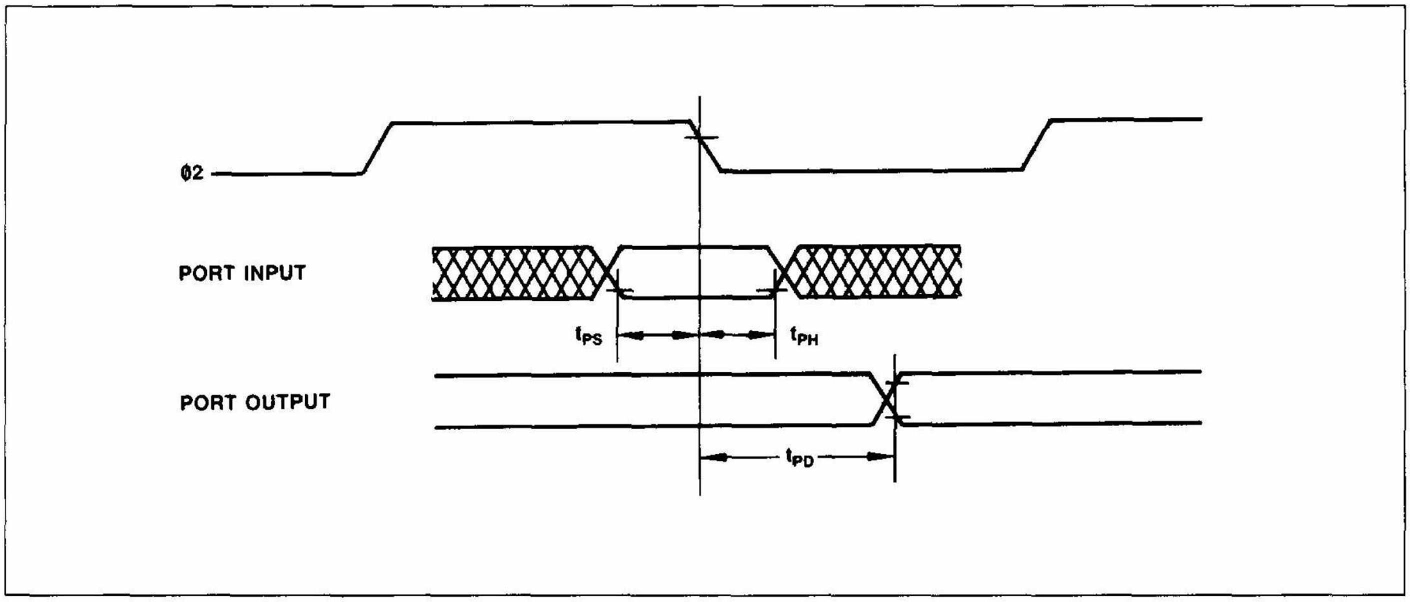
	0	
0	BRK	—OP Code
	Implied	—Addressing Mode
	1 7	—Instruction Bytes; Machine Cycles

□ —New Opcode

*Add 1 to N if page boundary is crossed.

**Add 1 to N if branch occurs to same page;
Add 2 to N if branch occurs to different page.

I/O PORT WAVEFORMS—ALL PORTS



I/O PORT TIMING—ALL PORTS

		2 MHz	4 MHz		
Parameter	**Symbol**	**Min**	**Max**	**Min**	**Max**
Input Data Setup Time	t_{PS}	50	—	35	—
Input Data Hold Time (Port D)	t_{PH}	10	—	10	—
Input Data Hold Time (All ports except D)		25	—	25	—
Output Data Delay Time	t_{PD}	—	120	—	100

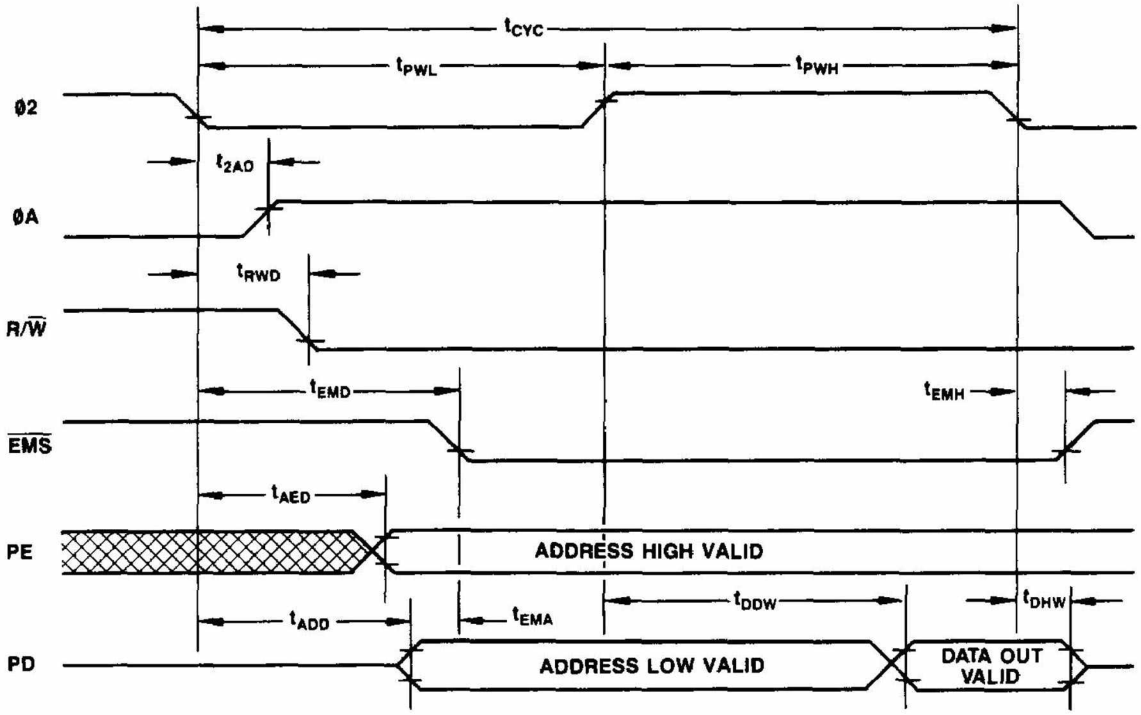
EXPANSION BUS TIMING

(V_{CC} = 5.0V ± 10%, T_A = 0°C to 70°C

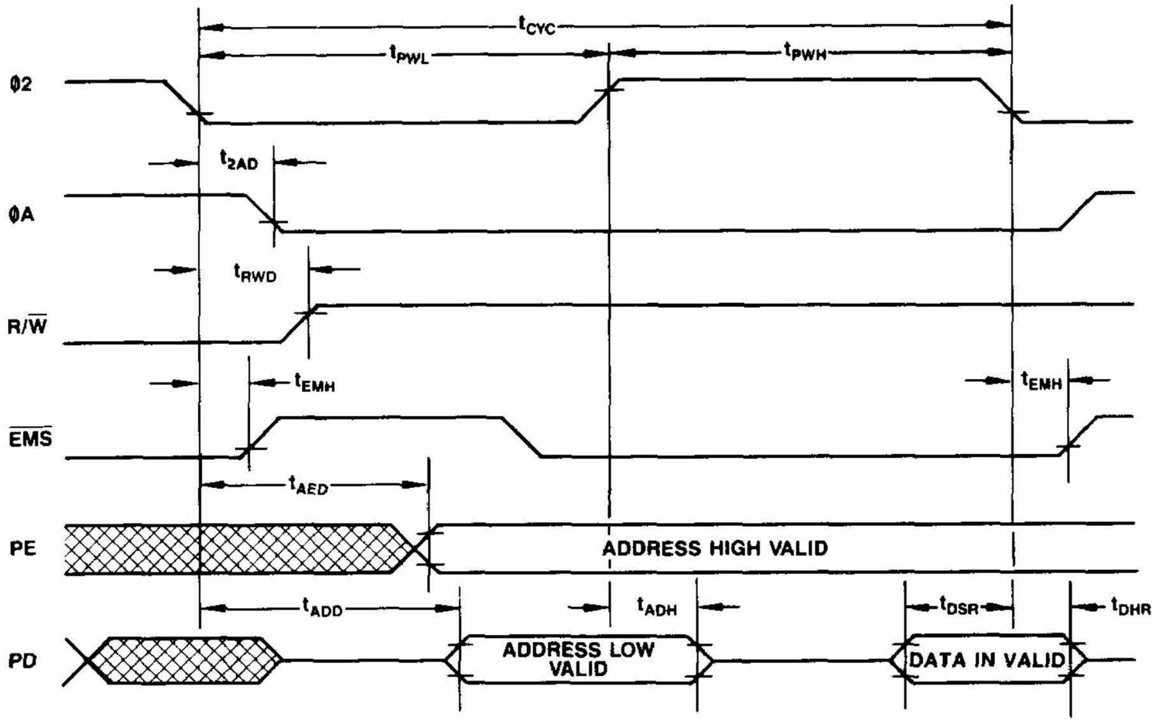
Parameter	Symbol	2 MHz		4 MHz		Unit
		Min.	Max.	Min.	Max.	
Clock/Control Timing						
ø2 Cycle Time	t_{CYC}	500	See Note 1	250	See Note 1	ns
Pulse Width ø2 Low	t_{PWL}	235	265	115	135	ns
Pulse Width ø2 High	t_{PWH}	235	265	115	135	ns
øA Delay Time — ø2 to ø2A	t_{2AD}	0	60	0	50	ns
$\overline{\text{EMS}}$ Delay Time — Address Valid to $\overline{\text{EMS}}$ Low	t_{EMA}	10	—	10	—	ns
$\overline{\text{EMS}}$ Delay Time — ø2 to $\overline{\text{EMS}}$ Low	t_{EMD}	—	150	—	115	ns
$\overline{\text{EMS}}$ Hold Time	t_{EMH}	10	—	10	—	ns
$\text{R}\overline{\text{W}}$ Delay Time	t_{RWD}	20	100	10	80	ns
PE Address Delay Time	t_{AED}	20	100	10	80	ns
PD Address Delay Time	t_{ADD}	20	120	10	100	ns
Write Timing						
Data Delay Time — Write	t_{DDW}	—	120	—	100	ns
Data Hold Time — Write	t_{DHW}	20	—	20	—	ns
Read Timing						
PD Address Hold Time — Read	t_{ADH}	0	80	0	60	ns
Data Setup Time — Read	t_{DSR}	50	—	35	—	ns
Data Hold Time — Read	t_{DHR}	10	—	10	—	ns

Note: 1. The ø2 clock should never be held static at a dc level. The maximum cycle time (t_{CYC}) that guarantees no data loss to internal registers is 20 microseconds.

EXPANSION BUS WRITE CYCLE WAVE FORMS



EXPANSION BUS READ CYCLE WAVE FORMS



MAXIMUM RATINGS*

Parameter	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	Vdc
Input Voltage	V_{IN}	−0.3 to V_{CC} +0.3	Vdc
Operating Temperature	T_A	0 to +70	°C
Storage Temperature	T_{STG}	−55 to +150	°C

***NOTE:** This device contains circuitry to protect the inputs against damage due to high static voltages, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this circuit.

DC CHARACTERISTICS

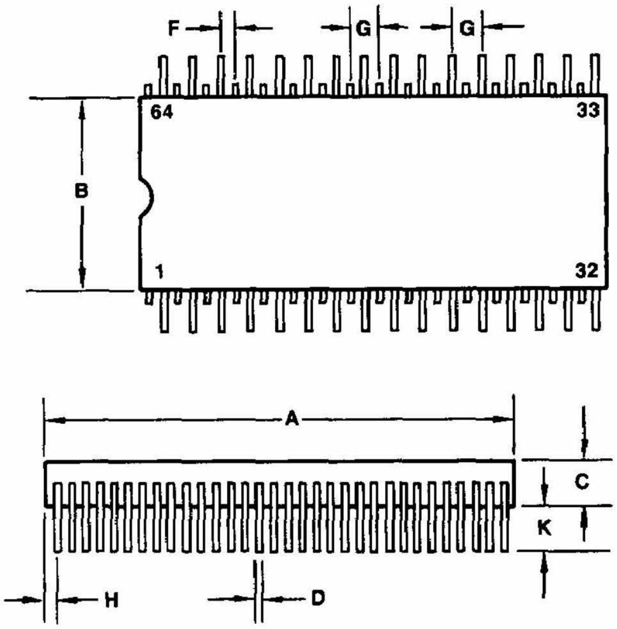
V_{CC} = +5.0V ± 10%, T_A = 0°C to 70°C (unless otherwise specified)

Parameter	Symbol	Min	Max	Unit	Test Condition
Input High Voltage	V_{IH}	+2.0	—	V	
Input Low Voltage	V_{IL}	—	+0.8	V	
Output High Voltage	V_{OH}	+2.4	—	V	V_{CC} = 4.5V I_{LOAD} = −100μA
Input Leakage Current	I_{IN}	—	±10	μA	V_{IN} = 0V or V_{CC} V_{CC} = 0V
Output Low Voltage	V_{OL}	—	+0.4	V	V_{CC} = 4.5V I_{LOAD} = 1.6 mA
Output Low Current (All ports except Port C)	I_{OUT}	—	−1.6	mA	V_{OL} = 0.4V
Input Capacitance (XTALO, XTALI) (All Others)	C_{IN}	— —	 25 5	 pF pF	V_{CC} = 5V I = 2 MHz T_A = 25°C
Output Capacitance	C_{OUT}	—	10	pF	
Operating Frequency Crystal or Master Clock Ø2 Clock	—	 .02 .01	 8.0 4.0	 MHz MHz	
Power Dissipation	P_D	—	40	mW	V_{CC} = 5V f = 2 MHz T_A = 25°C

Note: Negative sign indicates outward current flow, positive sign indicates inward current flow.

PACKAGE DIMENSIONS

64 PIN PLASTIC QUIP (QUAD IN-LINE PACKAGE)



DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	41.15	41.66	1.620	1.640
B	17.02	17.53	0.670	0.690
C	3.05	4.57	0.120	0.180
D	0.38	0.51	0.024	0.020
F	1.27 BSC		0.050 BSC	
G	2.54 BSC		0.100 BSC	
H	1.02	1.14	0.040	0.045
J	—	7°	—	7°
K	2.79	4.32	0.110	0.170
L	18.92	19.81	0.745	0.755
M	23.37	23.62	0.920	0.930