National Semiconductor

April 1992

COP404C ROMless CMOS Microcontrollers

General Description

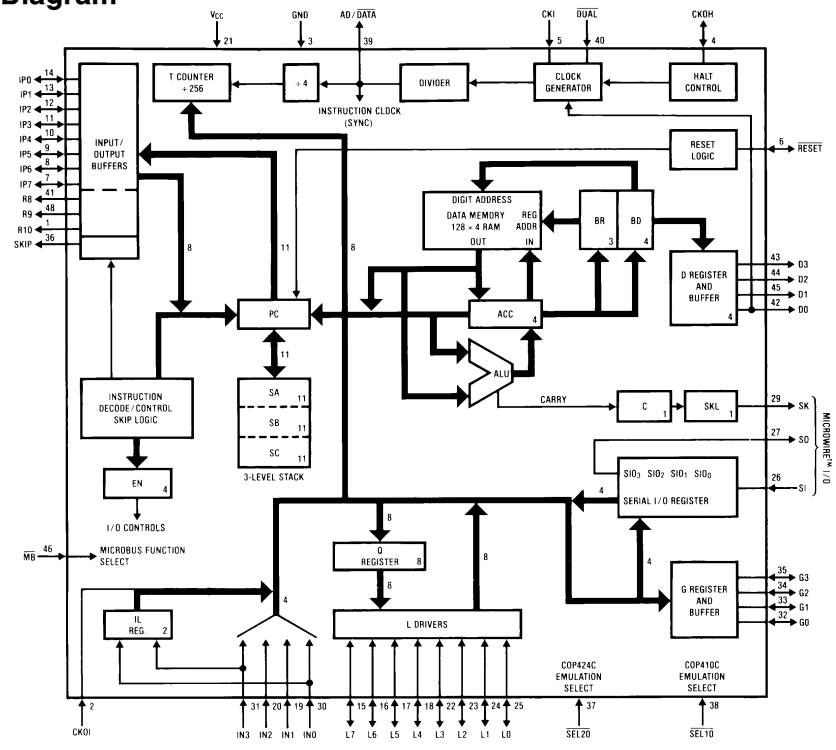
The COP404C ROMless Microcontroller is a member of the COPS™ family, fabricated using double-poly, silicon gate CMOS (microCMOS) technology. The COP404C contains CPU, RAM, I/O and is identical to a COP444C device except the ROM has been removed and pins have been added to output the ROM address and to input the ROM data. The COP404C can be configured, by means of external pins, to function as a COP444C, a COP424C, or a COP410C. Pins have been added to allow the user to select the various functional options that are available on the family of mask-programmed CMOS parts. The COP404C is primarily intended for use in the development and debug of a COP program for the COP444C/445C, COP424C/425C, and COP410C/411C devices prior to masking the final part. The COP404C is also appropriate in low volume applications or when the program might be changing.

Features

- Accurate emulation of the COP444C, COP424C and COP410C
- Lowest Power Dissipation (50 μW typical)
- Fully static (can turn off the clock)
- Power saving IDLE state and HALT mode
- 4 μs instruction time, plus software selectable clocks
- 128 × 4 RAM, addresses 2k × 8 ROM
- True vectored interrupt, plus restart
- Three-level subroutine stack
- Single supply operation (2.4V to 5.5V)
- Programmable read/write 8-bit timer/event counter
- Internal binary counter register with MICROWIRE™ serial I/O capability
- General purpose and TRI-STATE® outputs
- LSTTL/CMOS compatible
- MICROBUS™ compatible
- Software/hardware compatible with other members of the COP400 family

MICROBUS™ and MICROWIRE™ are trademarks of National Semiconductor Corporation.
TRI-STATE® is a registered trademark of National Semiconductor Corporation.

Block Diagram



TL/DD/5530–1

FIGURE 1. Block Diagram

  RRD-B30M105/Printed in U. S. A.

Absolute Maximum Ratings

Supply Voltage	6V	Operating temperature range	0° to +70°C
Voltage at any pin	−0.3V to V_{CC} + 0.3V	Storage temperature range	−65°C to +150°C
Total Allowable Source Current	25 mA	Lead temperature (soldering, 10 sec.)	300°C
Total Allowable Sink Current	25 mA		

DC Electrical Characteristics $0°C \leq T_a \leq 70°C$ unless otherwise specified

Parameter	Conditions	Min	Max	Units
Operating Voltage		2.4	5.5	V
Power Supply Ripple (Notes 4, 5)	peak to peak		0.1 V_{CC}	V
Supply Current (Note 1)	$V_{CC}=2.4V$, $t_c=64\ \mu s$		120	μA
	$V_{CC}=5.0V$, $t_c=16\ \mu s$		700	μA
	$V_{CC}=5.0V$, $t_c=4\ \mu s$ (T_c is instruction cycle time)		3000	μA
HALT Mode Current (Note 2)	$V_{CC}=5.0V$, $F_{IN}=0$ kHz, $T_A=25°C$		20	μA
	$V_{CC}=2.4V$, $F_{IN}=0$ kHz, $T_A=25°C$		6	μA
Input Voltage Levels				
$\overline{RESET}$, D0 (clock input)				
CKI				
Logic High		0.9 V_{CC}		V
Logic Low			0.1 V_{CC}	V
All other inputs (Note 7)				
Logic High		0.7 V_{CC}		V
Logic Low			0.2 V_{CC}	V
Input Pull-up current	$V_{CC}=4.5V$, $V_{IN}=0$	30	330	μA
Hi-Z input leakage		−1	+1	μA
Input capacitance (Note 4)			7	pF
Output Voltage Levels	Standard outputs			
LSTTL Operation	$V_{CC}=5.0V \pm 10\%$			
Logic High	$I_{OH}=-100\ \mu A$	2.7		V
Logic Low	$I_{OL}=400\ \mu A$		0.4	V
CMOS Operation				
Logic High	$I_{OH}=-10\ \mu A$	$V_{CC}-0.2$		V
Logic Low	$I_{OL}=10\ \mu A$		0.2	V
Output current levels				
Sink (Note 6)	$V_{CC}=4.5V$, $V_{OUT}=V_{CC}$	1.2		mA
	$V_{CC}=2.4V$, $V_{OUT}=V_{CC}$	0.2		mA
Source (Standard option)	$V_{CC}=4.5V$, $V_{OUT}=0V$	0.5		mA
	$V_{CC}=2.4V$, $V_{OUT}=0V$	0.1		mA
Source (Low current option)	$V_{CC}=4.5V$, $V_{OUT}=0V$	30	330	μA
	$V_{CC}=2.4V$, $V_{OUT}=0V$	6	80	μA
Allowable Sink/Source current per pin (Note 6)			5	mA
Allowable Loading on CKOH			100	pF
Current needed to over-ride HALT (Note 3)				
To continue	$V_{CC}=4.5V$, $V_{IN}=2V_{CC}$		.7	mA
To halt	$V_{CC}=4.5V$, $V_{IN}=7V_{CC}$		1.6	mA
TRI-STATE leakage current		−2.5	+2.5	μA

Note: Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.

COP404C

AC Electrical Characteristics $0°C \le T_A \le 70°C$ unless otherwise specified

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time (t_c)	$V_{CC} \ge 4.5V$ $4.5V > V_{CC} \ge 2.4V$	4 16	DC DC	μs μs
Operating CKI Frequency	$V_{CC} \ge 4.5V$ $4.5V > V_{CC} \ge 2.4V$	DC DC	1.0 250	MHz kHz
Duty Cycle (Note 4)	$f_1 = 4$ MHz	40	60	%
Rise Time (Note 4) Fall Time (Note 4)	$f_1 = 4$ MHz external clock		60 40	ns ns
Instruction Cycle Time using D0 as a RC Oscillator Dual-Clock Input (Note 4)	$R = 30k$, $V_{CC} = 5V$ $C = 82$ pF	8	16	μs
INPUTS: (See *Fig. 3*)				
t_{SETUP}	G Inputs SI Input IP Input All Others ($V_{CC} \ge 4.5V$)	$T_c/4 + .7$ 0.3 1.0 1.7		μs μs μs μs
t_{HOLD}	$V_{CC} \ge 4.5V$ $4.5V > V_{CC} \ge 2.4V$	0.25 1.0		μs μs
OUTPUT PROPAGATION DELAY	$V_{OUT} = 1.5V$, $C_L = 100$ pF, $R_L = 5K$			
IP7–IP0, A10–A8, SKIP t_{PD1}, t_{PD0}	$V_{CC} \ge 4.5V$ $4.5V > V_{CC} \ge 2.4V$		1.94 7.75	μs μs
AD/$\overline{DATA}$ t_{PD1}, t_{PD0}	$V_{CC} \ge 4.5V$ $4.5V > V_{CC} \ge 2.4V$		375 1.5	ns μs
ALL OTHER OUTPUTS t_{PD1}, t_{PD0}	$V_{CC} > 4.5V$ $4.5V > V_{CC} \ge 2.4V$		1.0 4.0	μs μs
MICROBUS TIMING Read Operation (*Fig. 4*)	$C_L = 50$ pF, $V_{CC} = 5V \pm 5\%$			
Chip select stable before $\overline{RD}$ $-t_{CSR}$		65		ns
Chip select hold time for $\overline{RD}$ $-t_{RCS}$		20		ns
$\overline{RD}$ pulse width $-t_{RR}$		400		ns
Data delay from $\overline{RD}$ $-t_{RD}$			375	ns
$\overline{RD}$ to data floating $-t_{DF}$ (Note 4)			250	ns
Write Operation (*Fig. 5*)				
Chip select stable before $\overline{WR}$ $-t_{CSW}$		65		ns
Chip select hold time for $\overline{WR}$ $-t_{WCS}$		20		ns
$\overline{WR}$ pulse width $-t_{WW}$		400		ns
Data set-up time for $\overline{WR}$ $-t_{DW}$		320		ns
Data hold time for $\overline{WR}$ $-t_{WD}$		100		ns
INTR transition time from $\overline{WR}$ $-t_{WI}$			700	ns

Note 1: Supply current is measured after running for 2000 cycle times with a square-wave clock on CKI and all other pins pulled up to V_{CC} with 20k resistors. See current drain equation on page 16.

Note 2: Test conditions: All inputs tied to V_{CC}; L lines in TRI-STATE mode and tied to Ground; all outputs tied to Ground.

Note 3: When forcing HALT, current is only needed for a short time (approx. 200 ns) to flip the HALT flip-flop.

Note 4: This parameter is only sampled and not 100% tested. Variation due to the device included.

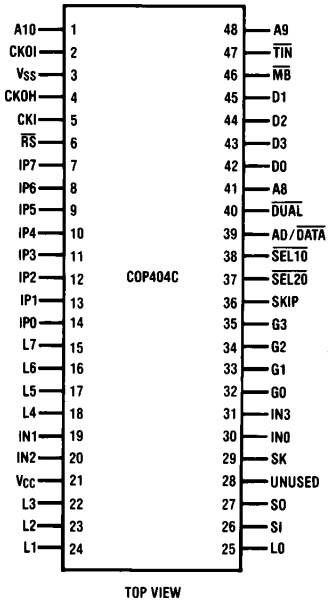
Note 5: Voltage change must be less than 0.1 V_{CC} in a 1 ms period.

Note 6: SO output sink current must be limited to keep V_{OL} less than 0.2 V_{CC} to prevent entering test mode.

Note 7: $\overline{MB}$, $\overline{TIN}$, $\overline{DUAL}$, $\overline{SEL10}$, $\overline{SEL20}$, input levels at V_{CC} or V_{SS}.

Connection Diagram

Dual-In-Line Package



TL/DD/5530–2

Order Number COP404CN
See NS Package Number N48A

Pin Descriptions

Pin	Description
V_{CC}	Most positive voltage
V_{SS}	Ground
CKI	Clock input
$\overline{RS}$	Reset input
CKOI	General purpose input
L0–L7	8 TRI-STATE I/O
G0–G3	4 general purpose I/O
D1–D3	3 general purpose outputs
D0	Either general purpose output or Dual-Clock RC input
IN0–IN3	4 general purpose inputs
SO	Serial data output
SI	Serial data input
SK	Serial data clock output
IP0–IP7	I/O for ROM address and data
A8, A9, A10	3 address outputs
SKIP	Skip status output
AD/$\overline{DATA}$	Clock output
$\overline{MB}$	MICROBUS select input
CKOH	Halt I/O pin
$\overline{DUAL}$	Dual-Clock select input
$\overline{TIN}$	Timer input select pin (should be connected to GND)
$\overline{SEL10}$	COP410C emulation select input
$\overline{SEL20}$	COP424C emulation select input
UNUSED	Ground

FIGURE 2

The internal architecture is shown in *Figure 1*. Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1", when a bit is reset, it is a logic "0".

PROGRAM MEMORY

Program Memory consists of a 2048-byte external memory (typically PROM). Words of this memory may be program instructions, constants or ROM addressing data.

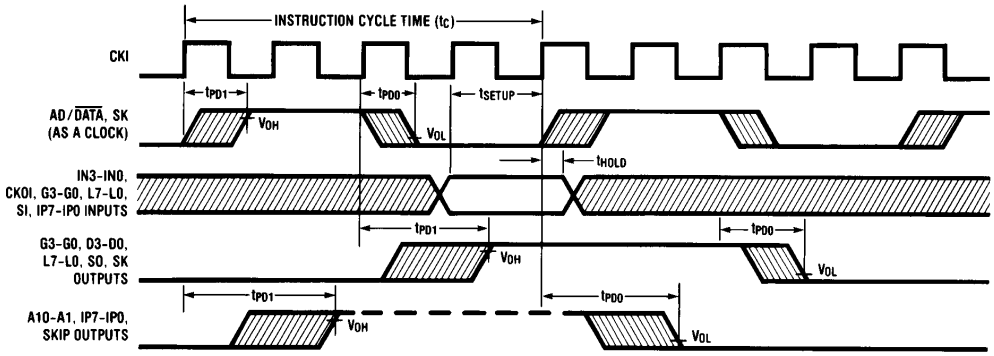
ROM addressing is accomplished by a 11-bit PC register which selects one of the 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 11-bit binary count value.

Three levels of subroutine nesting are implemented by a three level deep stack. Each subroutine call or interrupt pushes the next PC address into the stack. Each return pops the stack back into the PC register.

DATA MEMORY

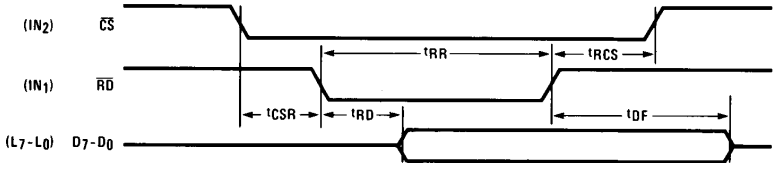
Data memory consists of a 512-bit RAM, organized as 8 data registers of 16 × 4-bit digits. RAM addressing is implemented by a 7-bit B register whose upper 3 bits (B_r) select 1 of 8 data registers and lower 4 bits (B_d) select 1 of 16 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) are usually loaded into or from, or exchanged with, the A register (accumulator), it may also be loaded into or from the Q latches or T counter or loaded from the L ports. RAM addressing may also be performed directly by the LDD and XAD instructions based upon the immediate operand field of these instructions. The B_d register also serves as a source register for 4-bit data sent directly to the D outputs.

Timing Diagrams



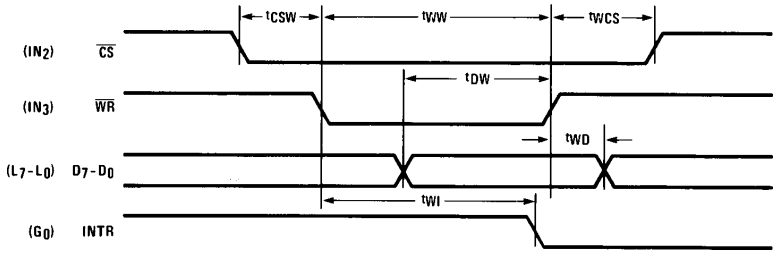
TL/DD/5530–3

FIGURE 3. Input/Output Timing



TL/DD/5530–4

FIGURE 4. MICROBUS Read Operation Timing



TL/DD/5530–5

FIGURE 5. MICROBUS Write Operation Timing

Functional Description

INTERNAL LOGIC

The processor contains its own 4-bit A register (accumulator) which is the source and destination register for most I/O, arithmetic, logic, and data memory access operations. It can also be used to load the B_r and B_d portions of the B register, to load and input 4 bits of the 8-bit Q latch or T counter, L I/O ports data, to input 4-bit G, or IN ports, and to perform data exchanges with the SIO register.

A 4-bit adder performs the arithmetic and logic functions, storing the results in A. It also outputs a carry bit to the 1-bit C register, most often employed to indicate arithmetic overflow. The C register in conjunction with the XAS instruction and the EN register, also serves to control the SK output.

The 8-bit T counter is a binary up counter which can be loaded to and from M and A using CAMT and CTMA instructions. This counter is operated as a time-base counter. When the T counter overflows, an overflow flag will be set (see SKT and IT instructions below). The T counter is cleared on reset. A functional block diagram of the timer/counter is illustrated in *Figure 10a*.

Four general-purpose inputs, IN3–IN0, are provided. IN1, IN2 and IN3 may be selected (by pulling $\overline{MB}$ pin low) as Read Strobe, Chip Select, and Write Strobe inputs, respectively, for use in MICROBUS application.

The D register provides 4 general-purpose outputs and is used as the destination register for the 4-bit contents of B_d. In the dual clock mode, D0 latch controls the clock selection (see dual oscillator below).

The G register contents are outputs to a 4-bit general-purpose bidirectional I/O port. G0 may be selected as an output for MICROBUS applications.

The Q register is an internal, latched, 8-bit register, used to hold data loaded to or from M and A, as well as 8-bit data from ROM. Its contents are outputted to the L I/O ports when the L drivers are enabled under program control. With the MICROBUS option selected, Q can also be loaded with the 8-bit contents of the L I/O ports upon the occurrence of a write strobe from the host CPU.

The 8 L drivers, when enabled, output the contents of latched Q data to the L I/O port. Also, the contents of L may be read directly into A and M. As explained above, the MICROBUS option allows L I/O port data to be latched into the Q register.

The SIO register functions as a 4-bit serial-in/serial-out shift register for MICROWIRE™ I/O and COPS peripherals, or as a binary counter (depending on the contents of the EN register). Its contents can be exchanged with A.

The XAS instruction copies C into the SKL latch. In the counter mode, SK is the output SKL; in the shift register mode, SK outputs SKL ANDed with the clock.

EN is an internal 4-bit register loaded by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register:

0. The least significant bit of the enable register, EN0, selects the SIO register as either a 4-bit shift register or a 4-bit binary counter. With EN0 set, SIO is an asynchronous binary counter, decrementing its value by one upon each low-going pulse (''1'' to ''0'') occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output equals the value of EN3. With EN0 reset, SIO is a serial shift register left shifting 1 bit each instruction cycle time. The data present at SI goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. The SK outputs SKL ANDed with the instruction cycle clock.
1. With EN1 set, interrupt is enabled. Immediately following an interrupt, EN1 is reset to disable further interrupts.
2. With EN2 set, the L drivers are enabled to output the data in Q to the L I/O port. Resetting EN2 disables the L drivers, placing the L I/O port in a high-impedance input state.
3. EN3, in conjunction with EN0, affects the SO output. With EN0 set (binary counter option selected) SO will output the value loaded into EN3. With EN0 reset (serial shift register option selected), setting EN3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN3 with the serial shift register option selected disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains set to ''0''.

INTERRUPT

The following features are associated with interrupt procedure and protocol and must be considered by the programmer when utilizing interrupts.

a. The interrupt, once recognized as explained below, pushes the next sequential program counter address (PC + 1) onto the stack. Any previous contents at the bottom of the stack are lost. The program counter is set to hex address 0FF (the last word of page 3) and EN1 is reset.

b. An interrupt will be recognized only on the following conditions:
   1. EN1 has been set.
   2. A low-going pulse (''1'' to ''0'') at least two instruction cycles wide has occurred on the IN1 input.
   3. A currently executing instruction has been completed.

TABLE I. ENABLE REGISTER MODES — BITS EN0 AND EN3

EN0	EN3	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL = 1, SK = clock If SKL = 0, SK = 0
0	1	Shift Register	Input to Shift Register	Serial out	If SKL = 1, SK = clock If SKL = 0, SK = 0
1	0	Binary Counter	Input to Counter	0	SK = SKL
1	1	Binary Counter	Input to Counter	1	SK = SKL

Functional Description (Continued)

4. All successive transfer of control instructions and successive LBIs have been completed (e.g. if the main program is executing a JP instruction which transfers program control to another JP instruction, the interrupt will not be acknowledged until the second JP instruction has been executed).

c. Upon acknowledgement of an interrupt, the skip logic status is saved and later restored upon popping of the stack. For example, if an interrupt occurs during the execution of an ASC (Add with Carry, Skip on Carry) instruction which results in carry, the skip logic status is saved and program control is transferred to the interrupt servicing routine at hex address 0FF. At the end of the interrupt routine, a RET instruction is executed to pop the stack and return program control to the instruction following the original ASC. At this time, the skip logic is enabled and skips this instruction because of the previous ASC carry. Subroutines should not be nested within the interrupt service routine, since their popping of the stack will enable any previously saved main program skips, interfering with the orderly execution of the interrupt routine.

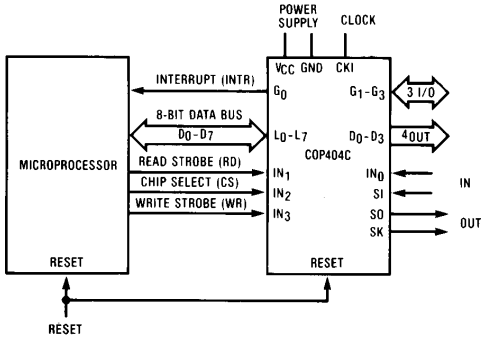
d. The instruction at hex address 0FF must be a NOP.

e. An LEI instruction may be put immediately before the RET instruction to re-enable interrupts.

MICROBUS INTERFACE

With $\overline{MB}$ pin tied to Ground, the COP404C can be used as a peripheral microprocessor device, inputting and outputting data from and to a host microprocessor (μP). IN1, IN2 and IN3 general purpose inputs become MICROBUS compatible read-strobe, chip-select, and write-strobe lines, respectively. IN1 becomes $\overline{RD}$ — a logic ''0'' on this input will cause Q latch data to be enabled to the L ports for input to the μP. IN2 becomes $\overline{CS}$ — a logic ''0'' on this line selects the COP404C and the μP peripheral device by enabling the operation of the $\overline{RD}$ and $\overline{WR}$ lines and allows for the selection of one of several peripheral components. IN3 becomes $\overline{WR}$ — a logic ''0'' on this line will write bus data from the L ports to the Q latches for input to the COP404C. G0 becomes INTR a ''ready'' output, reset by a write pulse from the μP on the $\overline{WR}$ line, providing the ''handshaking'' capability necessary for asynchronous data transfer between the host CPU and the COP404C.

This option has been designed for compatibility with National's MICROBUS - a standard interconnect system for 8-bit parallel data transfer between MOS/LSI CPUs and interfacing devices. (See MICROBUS National Publication). The functioning and timing relationships between the signal lines affected by this option are as specified for the MICROBUS interface, and are given in the AC electrical characteristics and shown in the timing diagrams (*Figures 4* and *5*). Connection of the COP404C to the MICROBUS is shown in *Figure 6*.



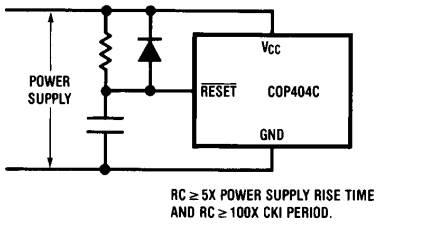
TL/DD/5530–7

FIGURE 6. MICROBUS Option Interconnect

INITIALIZATION

The external RC network shown in *Figure 7* must be connected to the $\overline{RESET}$ pin for the internal reset logic to initialize the device upon power-up. The $\overline{RESET}$ pin is configured as a Schmitt trigger input. If not used, it should be connected to V_{CC}. Initialization will occur whenever a logic ''0'' is applied to the $\overline{RESET}$ input, providing it stays low for at least three instruction cycle times.

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN, IL, T and G registers are cleared. The SKL latch is set, thus enabling SK as a clock output. Data Memory (RAM) is not cleared upon initialization. The first instruction at address 0 must be a CLRA (clear A register).



TL/DD/5530–8

FIGURE 7. Power-Up Circuit

TIMER

The timer is operated as a time-base counter. The instruction cycle frequency generated from CKI passes through a 2-bit divide-by-4 prescaler. The output of this prescaler increments the 8-bit T counter thus providing a 10-bit timer. The prescaler is cleared during execution of a CAMT instruction and on reset. For example, using a 1MHz crystal, the instruction cycle frequency of 250 kHz (divide by 4) increments the 10-bit timer every 4 μS. By presetting the counter and detecting overflow, accurate timeouts between 16 μS (4 counts) and 4.096 mS (1024 counts) are possible. Longer timeouts can be achieved by accumulating, under software control, multiple overflows.

HALT MODE

The COP404C is a FULLY STATIC circuit; therefore, the user may stop the system oscillator at any time to halt the chip. The chip may also be halted by two other ways (see *Figure 8*):

— Software HALT: by using the HALT instruction.

— Hardware HALT: by using the HALT I/O port CKOH. It is an I/O flip-flop which is an indicator of the HALT status. An external signal can over-ride this pin to start and stop the chip. By forcing CKOH high the

Functional Description (Continued)

chip will stop as soon as CKI is high and CKOH output will stay high to keep the chip stopped if the external driver returns to high impedance state.

Once in the HALT mode, the internal circuitry does not receive any clock signal and is therefore frozen in the exact state it was in when halted. All information is retained until continuing.

The chip may be awakened by one of two different methods:

— Continue function: by forcing CKOH low, the system clock will be re-enabled and the circuit will continue to operate from the point where it was stopped. CKOH will stay low.

— Restart: by forcing the $\overline{\text{RESET}}$ pin low (see Initialization)

The HALT mode is the minimum power dissipation state.

Note: if the user has selected dual-clock ($\overline{\text{DUAL}}$ pin tied to Ground) AND is forcing an external clock on D0 pin AND the COP404C is running from the D0 clock, the HALT mode - either hardware or software - will NOT be entered. Thus, the user should switch to the CKI clock to HALT. Alternatively, the user may stop the D0 clock to minimize power.

Oscillator Options

There are two basic clock oscillator configurations available as shown by *Figure 9*.

— CKI oscillator: CKI is configured as a LSTTL compatible input external clock signal. The external frequency is divided by 4 to give the instruction cycle time.

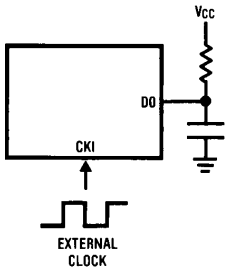
— Dual oscillator. By tying $\overline{\text{DUAL}}$ pin to Ground, pin D0 is now a single pin RC controlled Schmitt trigger oscillator input. The user may software select between the D0 oscillator (the instruction cycle time equals the D0 oscillation frequency divided by 4) by setting the D0 latch high or the CKI oscillator by resetting D0 latch low.

Note that even in dual clock mode, the counter, if used as a time-base counter, is always connected to the CKI oscillator.

For example, the user may connect up to a 1 MHz RC circuit to D0 for faster processing and a 32 kHz external clock to CKI for minimum current drain and time keeping.

Note: CTMA instruction is not allowed when the chip is running from D0 clock.

Figures 10a and *10b* show the timer and clock diagrams with and without Dual-Clock.



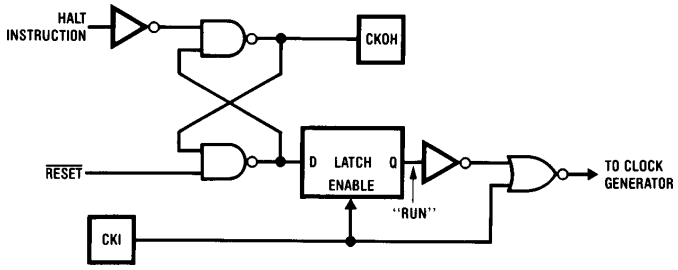
TL/DD/5530–9

R	C	Cycle Time	V_{CC}
15k	82 pF	4−9 μs	≥4.5V
30k	82 pF	8−16 μs	≥4.5V
60k	100 pF	16−32 μs	2.4−4.5V

Note: 15k≤R≤150k

50 pF≤C≤150 pF

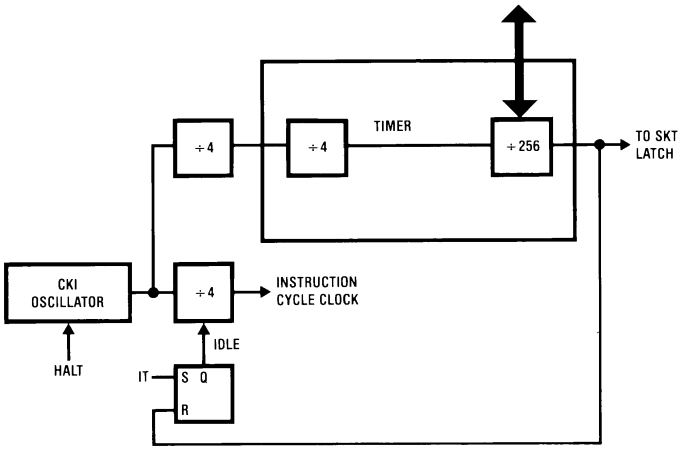
FIGURE 9. Dual-Oscillator Component Values



TL/DD/5530–10

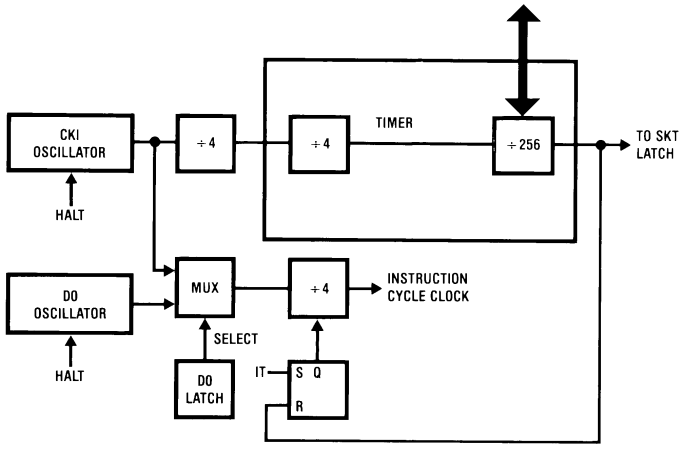
FIGURE 8. HALT Mode

Functional Description (Continued)



TL/DD/5530–11

FIGURE 10a. Clock and Timer Block Diagram without Dual-Clock



TL/DD/5530–12

Figure 10b. Clock and Timer Block Diagram with Dual-Clock

External Memory Interface

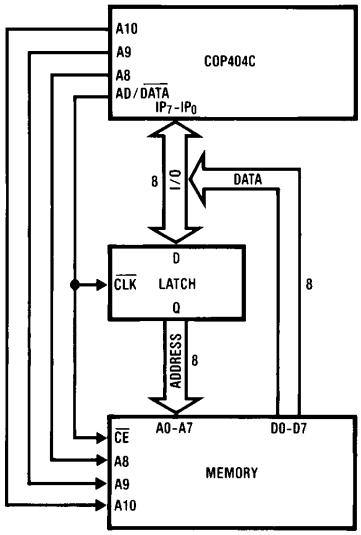
The COP404C is designed for use with an external Program Memory.

This memory may be implemented using any devices having the following characteristics:

1. random addressing
2. LSTTL or CMOS-compatible TRI-STATE outputs
3. LSTTL or CMOS-compatible inputs
4. access time = 1. 0 μs max.

Typically, these requirements are met using bipolar PROMs or MOS/CMOS PROMs, EPROMs or E^{2}PROMs.

During operation, the address of the next instruction is sent out on A10, A9, A8 and IP7 through IP0 during the time that AD/$\overline{\text{DATA}}$ is high (logic "1" = address mode). Address data on the IP lines is stored into an external latch on the high-to-low transition of the AD/DATA line; A10, A9 and A8 are dedicated address outputs, and do not need to be latched. When AD/$\overline{\text{DATA}}$ is low (logic "0" = data mode), the output of the memory is gated onto IP7 through IP0, forming the input bus. Note that AD/$\overline{\text{DATA}}$ output has a period of one instruction time, a duty cycle of approximately 50%, and specifies whether the IP lines are used for address output or data input. A simplified block diagram of the external memory interface is shown in *Figure 11.*



TL/DD/5530–13

FIGURE 11. External Memory Interface to COP404C

COP404C Instruction Set

Table II is a symbol table providing internal architecture, instruction operand and operation symbols used in the instruction set table.

Table III provides the mnemonic, operand, machine code data flow, skip conditions and description of each instruction.

Table II. Instruction Set Table Symbols

Symbol	Definition
Internal Architecture Symbols	
A	4-bit Accumulator
B	7-bit RAM address register
Br	Upper 3 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry register
D	4-bit Data output port
EN	4-bit Enable register
G	4-bit General purpose I/O port
IL	two 1-bit (IN0 and IN3) latches
IN	4-bit input port
L	8-bit TRI-STATE I/O port
M	4-bit contents of RAM addressed by B
PC	11-bit ROM address program counter
Q	8-bit latch for L port
SA	11-bit Subroutine Save Register A
SB	11-bit Subroutine Save Register B
SC	11-bit Subroutine Save Register C
SIO	4-bit Shift register and counter
SK	Logic-controlled clock output
SKL	1-bit latch for SK output
T	8-bit timer
Instruction operand symbols	
d	4-bit operand field, 0–15 binary (RAM digit select)
r	3-bit operand field, 0–7 binary (RAM register select)
a	11-bit operand field, 0–2047
y	4-bit operand field, 0–15 (immediate data)
RAM(x)	RAM addressed by variable x
ROM(x)	ROM addressed by variable x
Operational Symbols	
+	Plus
−	Minus
–>	Replaces
<–>	is exchanged with
=	Is equal to
−	
A	one's complement of A
⊕	exclusive-or
:	range of values

Instruction Set (Continued)

TABLE III. COP404C Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	|0011|0000|	A + C + RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	|0011|0001|	A + RAM(B) → A	None	Add RAM to A
ADT		4A	|0011|0001|	A + 10_{10} → A	None	Add Ten to A
AISC	y	5−	|0101| y |	A + y → A	Carry	Add Immediate. Skip on Carry (y ≠ 0)
CASC		10	|0001|0000|	$\overline{A}$ + RAM(B) + C → A Carry → C	Carry	Compliment and Add with Carry, Skip on Carry
CLRA		00	|0000|0000|	0 → A	None	Clear A
COMP		40	|0100|0000|	$\overline{A}$ → A	None	Ones complement of A to A
NOP		44	|0100|0100|	None	None	No Operation
RC		32	|0011|0010|	"0" → C	None	Reset C
SC		22	|0010|0010|	"1" → C	None	Set C
XOR		02	|0000|0010|	A ⊕ RAM(B) → A	None	Exclusive-OR RAM with A
TRANSFER OF CONTROL INSTRUCTIONS						
JID		FF	|1111|1111|	ROM ($PC_{10:8}$ A,M) → $PC_{7:0}$	None	Jump Indirect (note 2)
JMP	a	6−	|0110|0|$a_{10:8}$|	a → PC	None	Jump
		—	| $a_{7:0}$ |			
JP	a	—	|1| $a_{6:0}$ | (pages 2,3 only)	a → $PC_{6:0}$	None	Jump within Page (Note 3)
			or			
		—	|11| $a_{5:0}$ | (all other pages)	a → $PC_{5:0}$		
JSRP	a	—	|10| $a_{5:0}$ |	PC + 1 → SA → SB → SC 00010 → $PC_{10:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 4)
JSR	a	6−	|0110|1|$a_{10:8}$|	PC + 1 → SA → SB → SC a → PC	None	Jump to Subroutine
		—	| $a_{7:0}$ |			
RET		48	|0100|1000|	SC → SB → SA → PC	None	Return from Subroutine
RETSK		49	|0100|1001|	SC → SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
HALT		33	|0011|0011|		None	HALT processor
		38	|0011|1000|			
IT		33	|0011|0011|			IDLE till timer
		39	|0011|1001|		None	overflows then continues
MEMORY REFERENCE INSTRUCTIONS						
CAMT		33	|0011|0011|	A → $T_{7:4}$		
		3F	|0011|1111|	RAM(B) → $T_{3:0}$	None	Copy A, RAM to T
CTMA		33	|0011|0011|	$T_{7:44}$ → RAM(B)		
		2F	|0010|1111|	$T_{3:0}$ → A	None	Copy T to RAM, A
CAMQ		33	|0011|0011|	A → $Q_{7:4}$	None	Copy A, RAM to Q
		3C	|0011|1100|	RAM(B) → $Q_{3:0}$		
CQMA		33	|0011|0011|	$Q_{7:4}$ → RAM(B)	None	Copy Q to RAM, A
		2C	|0010|1100|	$Q_{3:0}$ → A		
LD	r	−5	|00| r |0101| (r = 0:3)	RAM(B) → A Br ⊕ r → Br	None	Load RAM into A, Exclusive-OR Br with r
LDD	r,d	23	|0010|0011|	RAM(r,d) → A	None	Load A with RAM pointed to direct by r,d
		—	|0| r | d |			
LQID		BF	|1011|1111|	ROM($PC_{10:8,A,M}$) → Q SB → SC	None	Load Q Indirect (Note 2)
RMB	0	4C	|0100|1100|	0 → $RAM(B)_0$	None	Reset RAM Bit
	1	45	|0100|0101|	0 → $RAM(B)_1$		
	2	42	|0100|0010|	0 → $RAM(B)_2$		
	3	43	|0100|0011|	0 → $RAM(B)_3$		

Instruction Set (Continued)

TABLE III. COP404C Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
SMB	0	4D	|0100|1101|	1 → $RAM(B)_0$	None	Set RAM Bit
	1	47	|0100|0111|	1 → $RAM(B)_1$		
	2	46	|0100|0110|	1 → $RAM(B)_2$		
	3	4B	|0100|1011|	1 → $RAM(B)_3$		
STII	y	7−	|0111| y |	y → RAM(B) Bd + 1 → Bd	None	Store Memory Immediate and Increment Bd
X	r	−6	|00| r |0110| (r=0:3)	RAM(B) ⟷ A Br ⊕ r → Br	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	r,d	23 —	|0010|0011| |1| r | d |	RAM(r,d) ⟷ A	None	Exchange A with RAM pointed to directly by r,d
XDS	r	−7	|00| r |0111| (r=0:3)	RAM(B) ⟷ A Bd−1 → Bd Br ⊕ r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd. Exclusive-OR Br with r
XIS	r	−4	|00| r |0100| (r=0:3)	RAM(B) ⟷ A Bd+1 → Bd Br ⊕ r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd, Exclusive-OR Br with r
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	|0101|0000|	A → Bd	None	Copy A to Bd
CBA		4E	|0100|1110|	Bd → A	None	Copy Bd to A
LBI	r,d	— or 33 — (any r, any d)	|00| r |(d-1)| (r=0:3: d=0,9:15) |0011|0011| |1| r | d |	r,d → B	Skip until not a LBI	Load B Immediate with r,d (Note 5)
LEI	y	33 6−	|0011|0011| |0110| y |	y → EN	None	Load EN Immediate (Note 6)
XABR		12	|0001|0010|	A ⟷ Br	None	Exchange A with Br (Note 7)
TEST INSTRUCTIONS						
SKC		20	|0010|0000|		C="1"	Skip if C is True
SKE		21	|0010|0001|		A=RAM(B)	Skip if A Equals RAM
SKGZ		33	|0011|0011|		$G_{3:0}=0$	Skip if G is Zero (all 4 bits)
		21	|0010|0001|			
SKGBZ		33	|0011|0011|	1st byte		Skip if G Bit is Zero
	0	01	|0000|0001|	2nd byte	$G_0=0$	
	1	11	|0001|0001|		$G_1=0$	
	2	03	|0000|0011|		$G_2=0$	
	3	13	|0001|0011|		$G_3=0$	
SKMBZ	0	01	|0000|0001|		$RAM(B)_0=0$	Skip if RAM Bit is Zero
	1	11	|0001|0001|		$RAM(B)_1=0$	
	2	03	|0000|0011|		$RAM(B)_2=0$	
	3	13	|0001|0011|		$RAM(B)_3=0$	
SKT		41	|0100|0001|		A time-base counter carry has occured since last test	Skip on Timer (Note 2)

Instruction Set (Continued)

TABLE III. COP404C Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
INPUT/OUTPUT INSTRUCTIONS						
ING		33	|0011|0011|	G ⟶ A	None	Input G Ports to A
		2A	|0010|1010|			
ININ		33	|0011|0011|	IN ⟶ A	None	Input IN Inputs to A
		28	|0010|1000|			
INIL		33	|0011|0011|	IL_3, CKO, ''0'', IL_0 ⟶ A	None	Input IL Latches to A (Note 2)
		29	|0010|1001|			
INL		33	|0011|0011|	$L_{7:4}$ ⟶ RAM(B)	None	Input L Ports to RAM,A
		2E	|0010|1110|	$L_{3:0}$ ⟶ A		
OBD		33	|0011|0011|	Bd ⟶ D	None	Output Bd to D Outputs
		3E	|0011|1110|			
OGI	y	33	|0011|0011|	y ⟶ G	None	Output to G Ports Immediate
		5—	|0101| y |			
OMG		33	|0011|0011|	RAM(B) ⟶ G	None	Output RAM to G Ports
		3A	|0011|1010|			
XAS		4F	|0100|1111|	A ⟷ SIO, C ⟶ SKL	None	Exchange A with SIO (Note 2)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicitly defined (e.g., Br and Bd are explicitly defined). Bits are numbered O to N where O signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit A register.

Note 2: For additional information on the operation of the XAS, JID, LQID, INIL, and SKT instructions, see below.

Note 3: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 4: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 5: LBI is a single-byte instruction if d = 0, 9, 10, 11, 12, 13, 14, or 15. The machine code for the lower 4 bits equals the binary value of the ''d'' data minus 1, e.g., to load the lower four bits of B(Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Note 6: Machine code for operand field y for LEI instruction should equal the binary value to be latched into EN, where a ''1'' or ''0'' in each bit of EN corresponds with the selection or deselection of a particular function associated with each bit. (See Functional Description, EN Register.)

Note 7: If $\overline{SEL2O}$ = 1, A ⟷ Br (0 ⟶ A3)
If $\overline{SEL2O}$ = 0, A ⟷ Br (0,0 ⟶ A3, A2).

Description of Selected Instructions

XAS INSTRUCTION

XAS (Exchange A with SIO) copies C to the SKL latch and exchanges the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. If SIO is selected as a shift register, an XAS instruction can be performed once every 4 instruction cycles to effect a continuous data stream.

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 11-bit word PC10: PC8, A, M. LQID can be used for table lookup or code conversion such as BCD to seven-segment. The LQID instruction ''pushes'' the stack (PC + 1 ⟶ SA ⟶ SB ⟶ SC) and replaces the least significant 8 bits of the PC as follows: A ⟶ PC (7:4), RAM(B) ⟶ PC(3:0), leaving PC(10), PC(9) and PC(8) unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is ''popped'' (SC ⟶ SB ⟶ SA ⟶ PC), restoring the saved value of PC to continue sequential program execution. Since LQID pushes SB ⟶ SC, the previous contents of SC are lost.

Note: LQID uses 2 instruction cycles if executed, one if skipped.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower 8 bits of the ROM address register PC with the contents of ROM addressed by the 11-bit word, PC10: 8, A, M. PC10, PC9 and PC8 are not affected by JID.

Note: JID uses 2 instruction cycles if executed, one if skipped.

Description of Selected Instructions (Continued)

SKT INSTRUCTION

The SKT (Skip On Timer) instruction tests the state of the T counter overflow latch (see internal logic, above), executing the next program instruction if the latch is not set. If the latch has been set since the previous test, the next program instruction is skipped and the latch is reset. The features associated with this instruction allow the processor to generate its own time-base for real-time processing, rather than relying on an external input signal

Note: If the most significant bit of the T counter is a 1 when a CAMT instruction loads the counter, the overflow flag will be set. The following sample of codes should be used when loading the counter:

```
CAMT    ; load T counter
SKT     ; skip if overflow flag is set and reset it
NOP
```

IT INSTRUCTION

The IT (idle till timer) instruction halts the processor and puts it in an idle state until the time-base counter overflows. This idle state reduces current drain since all logic (except the oscillator and time base counter) is stopped.

INIL INSTRUCTION

INIL (Input IL Latches to A) inputs 2 latches, IL3 and IL0, CKOI and 0 into A. The IL3 and IL0 latches are set if a low-going pulse (''1'' to ''0'') has occurred on the IN3 and IN0 inputs since the last INIL instruction, provided the input pulse stays low for at least two instruction cycles. Execution of an INIL inputs IL3 and IL0 into A3 and A0 respectively, and resets these latches to allow them to respond to subsequent low-going pulses on the IN3 and IN0 lines. The state of CKOI is input into A2. A 0 is input into A1. IL latches are cleared on reset.

Instruction Set Notes

a. The first word of a program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, they are still fetched from the program memory. Thus program paths take the same number of cycles whether instructions are skipped or executed except for JID, and LQID.

c. The ROM is organized into pages of 64 words each. The Program Counter is a 11-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID, or LQID is the last word of a page, it operates as if it were in the next page. For example: a JP located in the last word of a page will jump to a location in the next page. Also, a JID or LQID located in the last word of every fourth page (i.e. hex address 0FF, 1FF, 2FF, 3FF, 4FF, etc.) will access data in the next group of four pages.

Power Dissipation

The lowest power drain is when the clock is stopped. As the frequency increases so does current. Current is also lower at lower operating voltages. Therefore, for minimum power dissipation, the user should run at the lowest speed and voltage that his application will allow. The user should take care that all pins swing to full supply levels to insure that outputs are not loaded down and that inputs are not at some intermediate level which may draw current. Any input with a slow rise or fall time will draw additional current. For example, an RC oscillator on D0 will draw more current than a square wave clock input since it is a slow rising signal.

If using an external square wave oscillator, the following equation can be used to calculate the COP404C operating current drain:

$$I_{co} = Iq + V \times 40 \times F_i + V \times 1400 \times F_i / 4$$

where:

I_{co} = chip operating current drain in microamps

I_q = quiescent leakage current (from curve)

F_i = CKI frequency in MegaHertz

V = chip V_{CC} in volts

For example at 5 volts V_{CC} and 400 kHz:

$$I_{co} = 20 + 5 \times 40 \times .4 + 5 \times 1400 \times .4 / 4$$

$$I_{co} = 20 + 80 + 700 = 800\ \mu A$$

at 2.4 volts V_{CC} and 30 kHz:

$$I_{co} = 6 + 2.4 \times 40 \times .03 + 2.4 \times 1400 \times .0¾$$

$$I_{co} = 6 + 2.88 + 25.2 = 34.08\ \mu A$$

If an IT instruction is executed, the chip goes into the IDLE mode until the timer overflows. In IDLE mode, the current drain can be calculated from the following equation:

$$I_{ci} = I_q + V \times 40 \times F_i$$

For example, at 5 volts V_{CC} and 400 kHz

$$I_{ci} = 20 + 5 \times 40 \times .4 = 100\ \mu A$$

The total average current will then be the weighted average of the operating current and the idle current:

$$Ita = Ico \times \frac{To}{To + Ti} + Ici \times \frac{Ti}{To + Ti}$$

where:

I_{ta} = total average current

I_{co} = operating current

I_{ci} = idle current

T_o = operating time

T_i = idle time

I/O OPTIONS

COP404C outputs have the following configurations, illustrated in *Figure 12*.

a. Standard — A CMOS push-pull buffer with an N-channel device to ground in conjunction with a P-channel device to V_{CC}, compatible with CMOS and LSTTL. (Used on SO, SK, AD/$\overline{DATA}$, SKIP, A10:8 and D outputs.)

b. Low Current — This is the same configuration as a. above except that the sourcing current is much less. (Used on G outputs.)

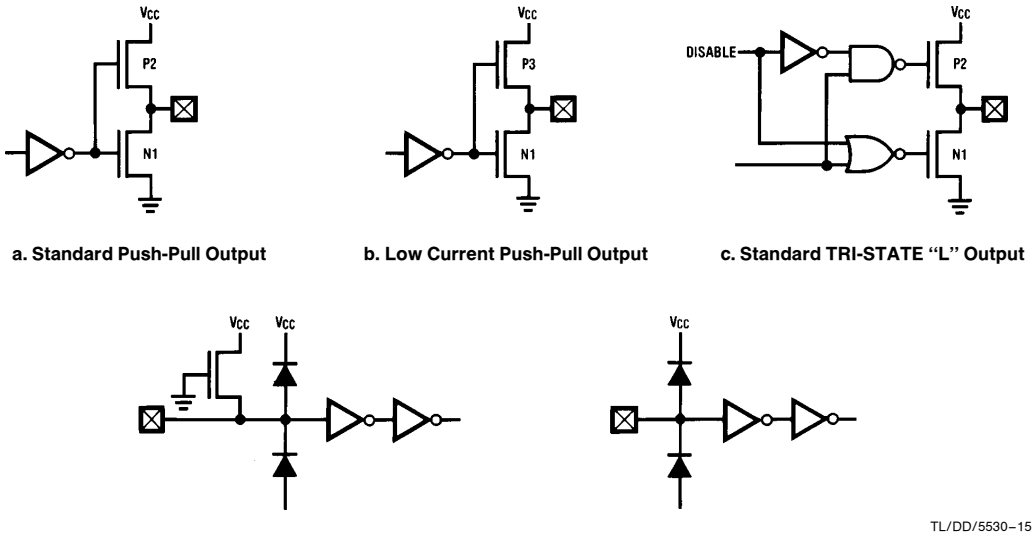
c. Standard TRI-STATE L Output — A CMOS output buffer similar to a. which may be disabled by program control. (Used on L outputs.)

All inputs have the following configuration:

d. Input with on chip load device to V_{CC}. (Used on CKOI.)

e. HI-Z input which must be driven by the users logic. (Used on CKI, $\overline{RESET}$, IN, SI, $\overline{DUAL}$, $\overline{MB}$, $\overline{SEL10}$ and $\overline{SEL20}$ inputs.)

All output drivers use one or more of three common devices numbered 1 to 3. Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figure 13* for each of these devices to allow the designer to effectively use these I/O configurations.



a. Standard Push-Pull Output

b. Low Current Push-Pull Output

c. Standard TRI-STATE "L" Output

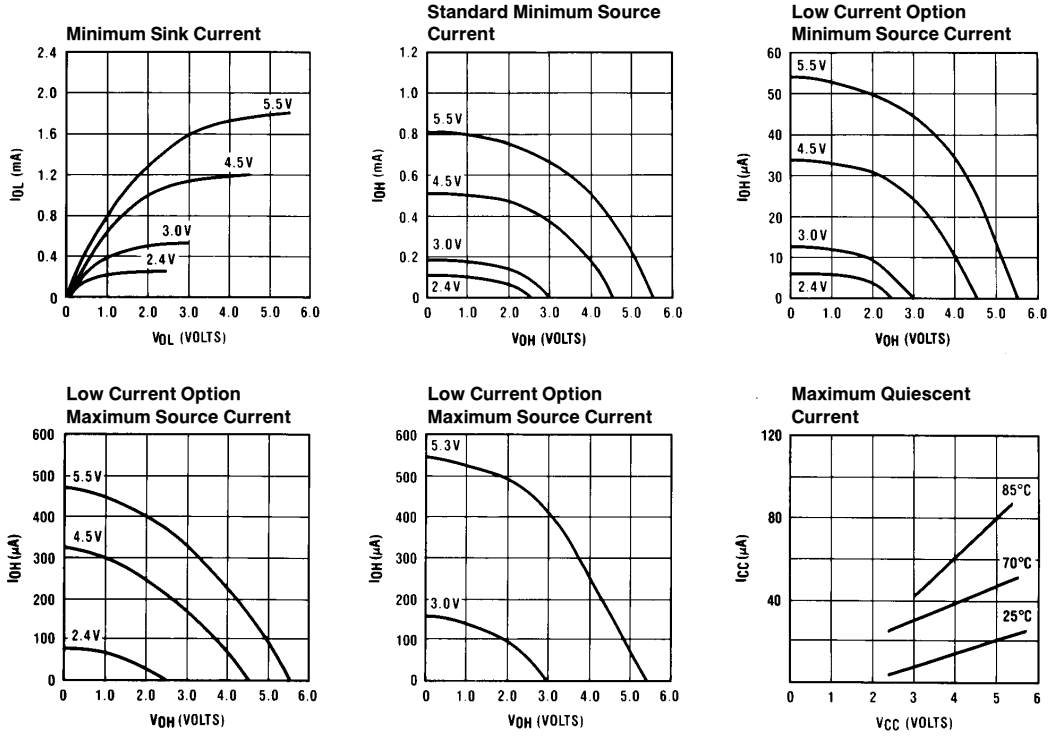
d. Input with Load

e. Hi-Z Input

TL/DD/5530–15

FIGURE 12. Input/Output Configurations

Typical Performance Characteristics



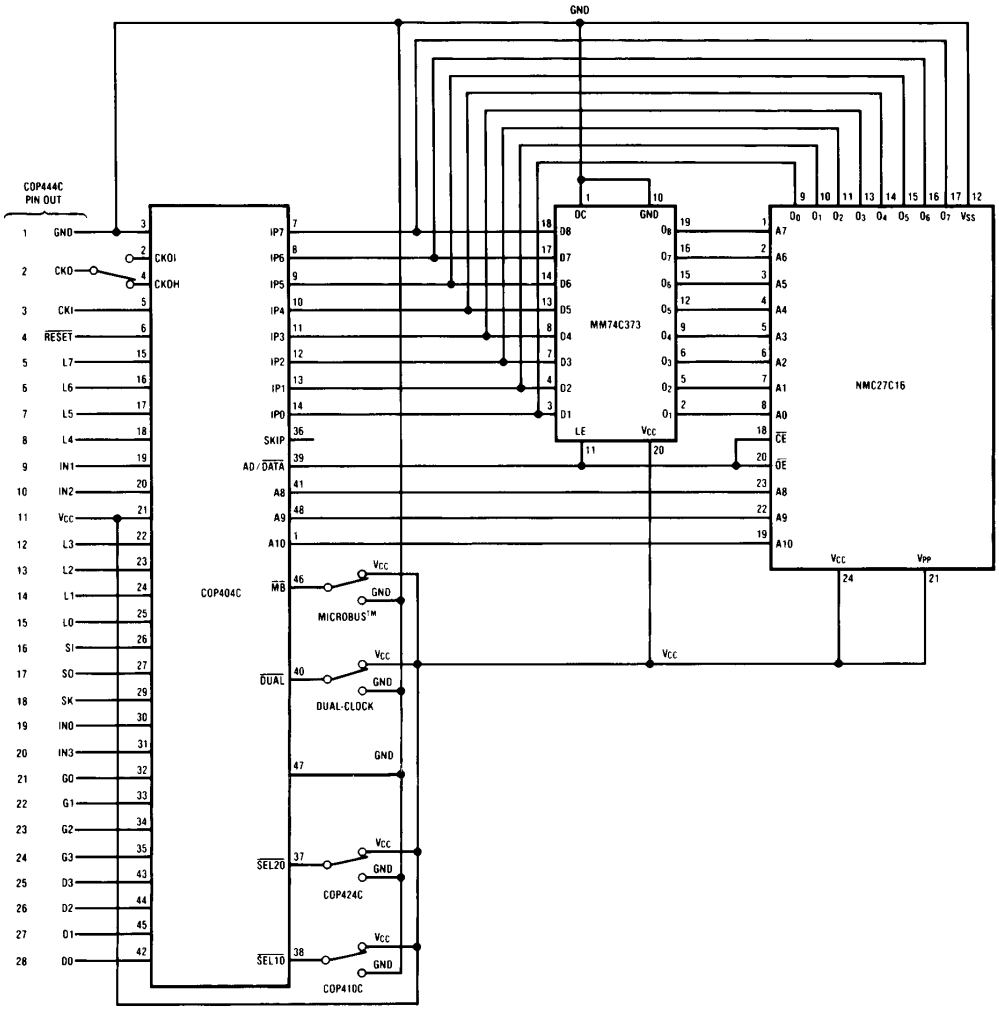
TL/DD/5530–16

FIGURE 13. Input/Output Characteristics

Emulation

The COP404C may be used to exactly emulate the COP444C/445C, COP424C/425C, and COP410C/411C. However, the Program Counter always addresses 2k of external ROM whatever chip is being emulated. *Figure 14* shows the interconnect to implement a hardware emulation. This connection uses a NMC27C16 EPROM as external memory. Other memory can be used such as bipolar PROM or RAM.

Pins IP7–IP0 are bidirectional inputs and outputs. When the AD/$\overline{\text{DATA}}$ clocking output turns on, the EPROM drivers are disabled and IP7–IP0 output addresses. The 8-bit latch (MM74C373) latches the addresses to drive the memory.



TL/DD/5530–14

FIGURE 14. COP404C Used To Emulate A COP444C

Emulation (Continued)

When AD/$\overline{DATA}$ turns off, the EPROM is enabled and the IP7–IP0 pins will input the memory data. A10, A9 and A8 output the most significant address bits to the memory. (SKIP output may be used for program debug if needed.)

— CKI is divided by 4. Other divide-by are emulated by external divider.

— CKO can be emulated as a general purpose input by using CKOI or as a Halt I/O port by using CKOH.

— $\overline{MB}$ pin can be pulled low if the MICROBUS feature of the COP444C and COP424C is needed. Othewise it should be high.

— $\overline{DUAL}$ pin can be pulled low if the Dual-Clock feature of the COP444C and COP424C is needed. Otherwise it should be high.

— The $\overline{SEL1O}$ and $\overline{SEL2O}$ inputs are used to emulate the COP444C/445C, COP424C/425C, or COP410C/411C.

- When emulating the COP444C/445C, the user must configure $\overline{SEL2O}$ = 1 and $\overline{SEL1O}$ = 1.
- When emulating the COP424C/425C, the user must configure $\overline{SEL2O}$ = 0 and $\overline{SEL1O}$ = 1. In this mode, the user RAM is physically halved. As in the COP424C/425C, the user has 64 digits (256 bits) of RAM available. Pin A10 should not be connected to the program memory (most significant address bit of the program memory should be grounded if using a 2k × 8 memory).
- When emulating the COP410C/411C, the user must configure $\overline{SEL2O}$ = 0 and $\overline{SEL1O}$ = 0. In this mode, the user has 32 digits (128 bits) of RAM available organized in the same way as the COP410C/411C - 4 registers of 8 digits each. Pins A10 and A9 should not be connected to the program memory (the 2 most significant address bits of the program memory should be grounded).

Furthermore, the subroutine stack is decreased from 3 levels to 2 levels.

The pins $\overline{SEL1O}$ and $\overline{SEL2O}$ change the internal logic of the device to accurately emulate the devices as indicated above. However, the user must remember that the COP424C/425C is a subset of the COP444C/COP445C with respect to memory size. The COP410C/411C is a subset both in memory size and in function. The user must take care not to use features and instructions which are not available on the COP410C/411C (see table IV. below) when using the COP404C to emulate the COP410C/411C.

TABLE IV. FEATURES AND INSTRUCTIONS NOT AVAILABLE ON COP410C/411C.

Timer	ADT		
Dual-clock	CASC		
Interrupt	CAMT		
Microbus	CTMA		
	IT		
	LDD	r, d	
	XAD	r, d	(except 3, 15)
	XABR		
	SKT		
	ININ		
	INIL		
	OGI	y	

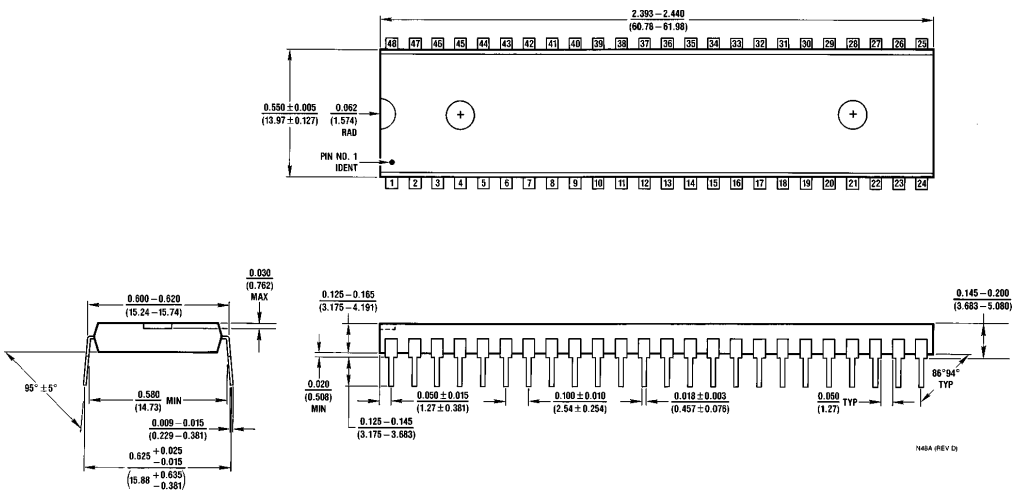
Option Table

COP404C MASK OPTIONS

The following COP444C options have been implemented in the COP404C:

Option value	Comment
Option 1 = 0	Ground Pin — no option available
Option 2 = 1, 2	CKO is replaced by CKOI and CKOH
Option 3 = 5	CKI is external clock input divided by 4
Option 4 = 1	$\overline{RESET}$ is Hi-Z input
Option 5–8 = 0	L outputs are standard TRI-STATE
Option 9 = 1	IN1 is a Hi-Z input
Option 10 = 1	IN2 is a Hi-Z input
Option 11 = 0	V_{CC} pin — no option available
Option 12–15 = 0	L outputs are standard TRI-STATE
Option 16 = 1	SI is a Hi-Z input
Option 17 = 0	SO is a standard output
Option 18 = 0	SK is a standard output
Option 19 = 1	IN0 is a Hi-Z input
Option 20 = 1	IN3 is a Hi-Z input
Option 21–24 = 1	G outputs are low-current
Option 25–28 = 0	D outputs are standard
Option 29 = 1	No internal initialization logic
Option 30 = 0, 1	DUAL-CLOCK is pin selectable
Option 31 = 0	TIMER time-base counter
Option 32 = 0, 1	MICROBUS is pin selectable
Option 33 = N/A	48-pin package

Physical Dimensions inches (millimeters)



Molded Dual-In-Line Package (N)
Order Number COP404CN
NS Package Number N48A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408