NEC microcomputers, inc.

μPD546

# μCOM-43
# SINGLE CHIP MICROCOMPUTER
## USERS' MANUAL

Five Militia Drive/Lexington, Massachusetts 02173 Telephone (617) 862-6410 Telex 92-3434 TWX 710-326-6520

# uCOM-43 SINGLE CHIP MICROCOMPUTER

# USERS' MANUAL

JANUARY 1978

NEC MICROCOMPUTERS, INC.

uCOM-43
NEC MOS DIGITAL INTEGRATED CIRCUIT
4-BIT SINGLE CHIP MICROCOMPUTER

## INTRODUCTION

The NEC uCOM-43 is a 4-bit parallel central processing unit that forms a single chip microcomputer especially suited for a broad range of low cost and sophisticated controller applications.

The uCOM-43 contains all the necessary system functional blocks for a single chip microcomputer, including a 4-bit parallel ALU, 2,000 by 8-bit program ROM, 96 by 4-bit data RAM, 35 input/output channels, a programmable interval timer, interrupt handling circuits, a clock generator and control circuits.

The instruction set of the uCOM-43 includes 80 powerful instructions. The instruction set features controller-oriented functions and efficient use of program memory, via a variety of multi-function instructions, powerful I/O instructions, and a number of bit manipulation and test-and-skip instructions.

The extensive flexibility and processing capabilities provided by the uCOM-43 will enable advanced microcomputerization of both industrial and non-industrial controller applications.

## FEATURES

- Single chip microcomputer for controller applications.
- 2,000 by 8-bit ROM for program storage.
- 96 by 4-bit RAM for data storage.
- 35 input/output channels,
  with single bit manipulation and 4-bit parallel processing capabilities for all ports.

  | | |
  |---|---|
  | Two 4-bit input ports | : A and B |
  | Two 4-bit input/output ports | : C and D |
  | Four 4-bit output ports | : E, F, G and H |
  | One 3-bit output port | : I |

- 3 level program counter stack,
  for 3 level subroutines or 2 level subroutines and an interrupt service.
- Six 4-bit working registers in a portion of RAM.

- An interrupt request input line,
  with interrupt enable/disable capability.

- Built-in 6-bit programmable interval timer,
  enabling 64 different time intervals or greater with use
  of RAM, and parallel processing to increase throughput.

- Built-in clock generator circuit,
  controlled with external, low-cost IFT (intermediate
  frequency transformer).

- Powerful 80 instruction set.
  73 single-byte instructions and 7 double-byte
  instructions.

- A variety of multi-function instructions to increase
  throughput.

- Powerful input/output instructions.

- Bit manipulation and test-and-skip instructions.

- Binary addition, decimal addition and subtraction, and
  logical operations.

- A variety of subroutine call instructions.

  1 byte call instruction for calling
  fixed addresses in page 0.

  2 byte call instruction for calling
  any address in ROM.

- Instruction cycle time -- 10usec.

- Open drain outputs.

- P-channel MOS.

- Single power supply, -10V.

- 42 pin plastic DIP.

- Two versions available:

  uPD546C -- Fully TTL compatible.

  uPD553C -- Outputs capable of -40 volts for direct interfacing to vacuum fluorescent displays.

- CMOS version available 2Q78.

- Development support tools.

  64 pin evaluation chip (uPD556D) with CPU and RAM on chip, for prototyping with external program memory.

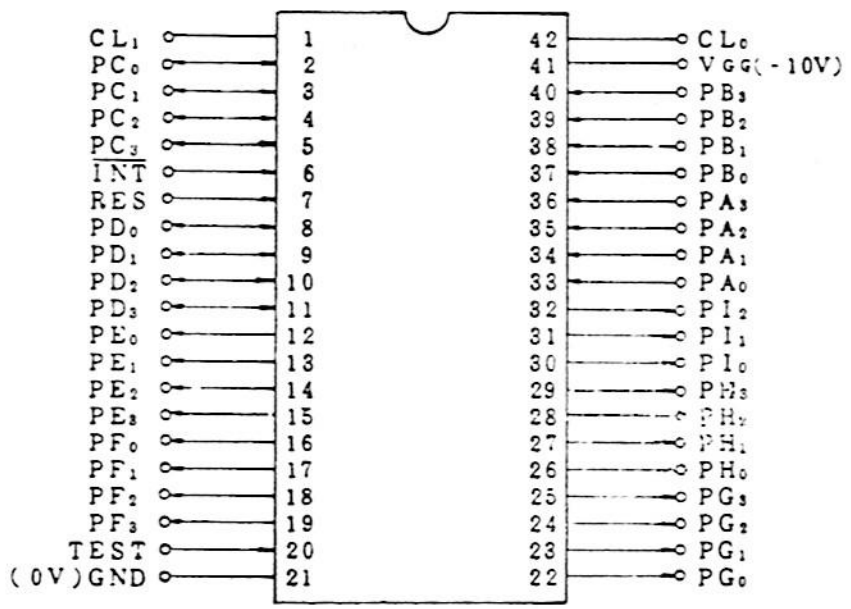
  Evaluation kit (EVAKIT-43) with evaluation chip and PROMs on board, including hardware-implemented system control and monitoring capability.

  Cross assembler on NEC PDA-80, the Program Development Aid microcomputer system based on the 8080A.

  FORTRAN IV cross assembler.

## PIN ASSIGNMENT

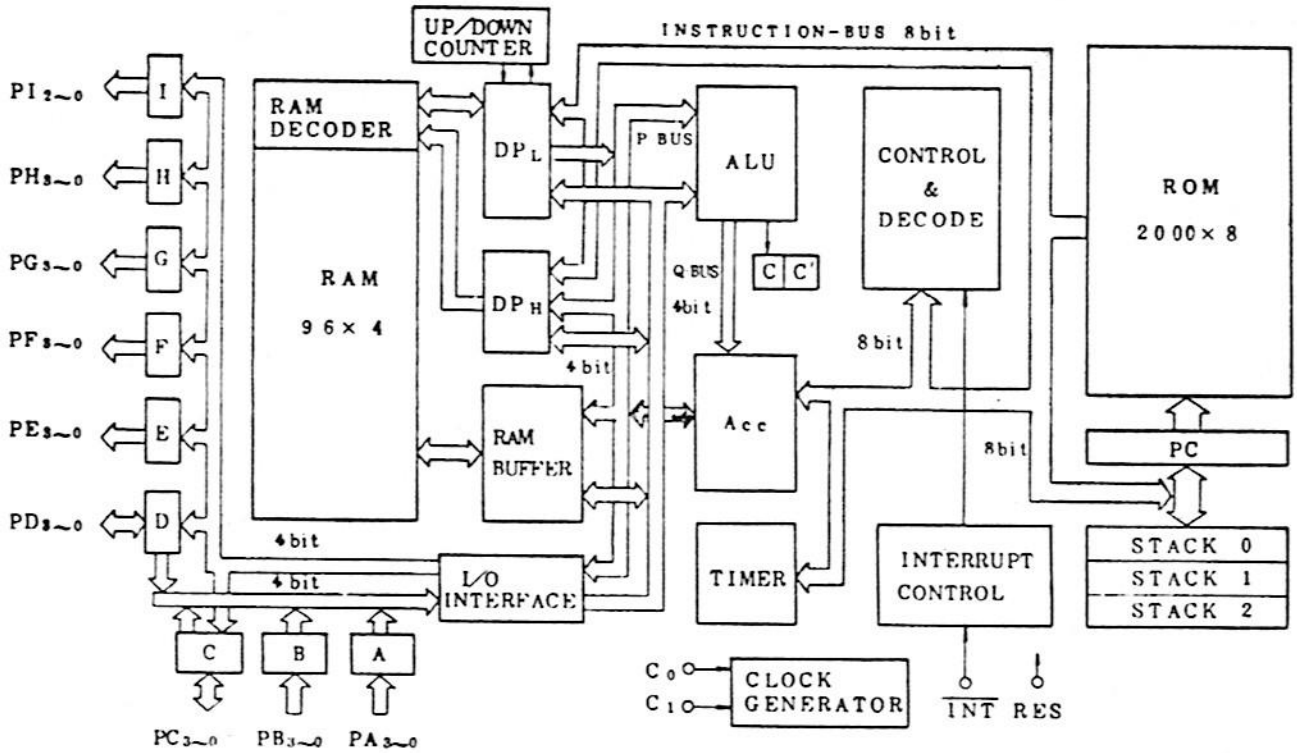
Pin Configuration (Top View)



Pin Names and Functions

| NAME | Pin No. | Input/Output | Functions |
|---|---|---|---|
| INT-NOT | 6 | Input | Interrupt request input. |
| RES | 7 | Input | System reset input. |
| PA3-0 | 36-33 | Input | Two 4-bit input ports A3-0 and B3-0, each capable of 4-bit parallel input and any single bit test for test-and-skip. |
| PB3-0 | 40-37 | | |
| PC3-0 | 5-2 | Input/Output | Two independent 4-bit input/output ports C3-0 and D3-0. As an input port, each is capable of 4-bit parallel input and any single bit test for test-and-skip operation. As an output port, each is capable of 4-bit parallel output. An 8-bit immediate data can also be output using both ports simultaneously. |
| PD3-0 | 11-8 | | |
| PE3-0 | 15-12 | Output | Four 4-bit output ports E3-0, F3-0, G3-0 and H3-0. Each port is capable of 4-bit parallel output and any single bit set/reset. |
| PF3-0 | 19-16 | | |
| PG3-0 | 25-22 | | |
| PH3-0 | 29-26 | | |
| PI2-0 | 32-30 | Output | 3-bit output port I2-0. Capable of 3-bit parallel output and any single bit set/reset. |
| CL1,0 | 1, 42 | -- | Connection for internal clock oscillation source, such as an IFT (intermediate frequency transformer), or an external clock input. |
| TEST | 20 | Input | Device testing terminal which is tied to GND(0V) in normal operation. |
| VGG | 41 | -- | -10V single power supply. |
| GND | 21 | -- | Ground 0V. |

BLOCK DIAGRAM

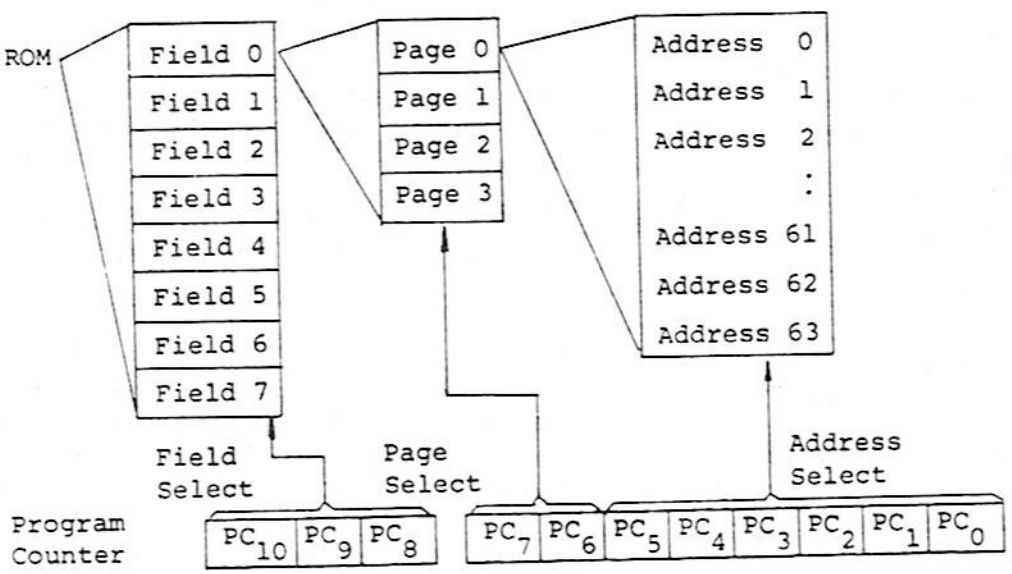


FUNCTIONAL BLOCKS

PROGRAM MEMORY (ROM)

The user's application program is stored in the 2,000 word by 8-bit mask programmable read only memory (ROM). The ROM is organized into fields and pages. The 2,000 word ROM is divided into 8 fields. Each field is subdivided into 4 pages of 64 words each, and each word consists of 8 bits. Since the ROM size totals 2,000 words, the last page (4th page in the 8th field) contains only 16 words. All the other 31 pages contain 64 words each. The ROM address range available to the user is 000 to 7CF in hexadecimal, the last address 7CF being located at field 7, page 3, address 15. The 11 bit program counter is used to address any of the 2,000 ROM locations.

ROM Organization and Program Counter




1 Field = 4 Pages
1 Page = 64 Addresses

PROGRAM COUNTER (PC)

The contents of the program counter point to a specific memory address in the 2,000 word ROM area in order to fetch the

next instruction to be executed. The 11-bit program counter is organized as a 3-bit register (higher 3 bits) and an 8-bit binary up counter (lower 8 bits). The contents in the 3-bit register specify one of 8 fields of the ROM. The 8-bit binary counter is divided so that the contents in the higher 2 bits specify one of four pages in a field and the contents in the lower 6 bits specify one of 64 addresses in a page.

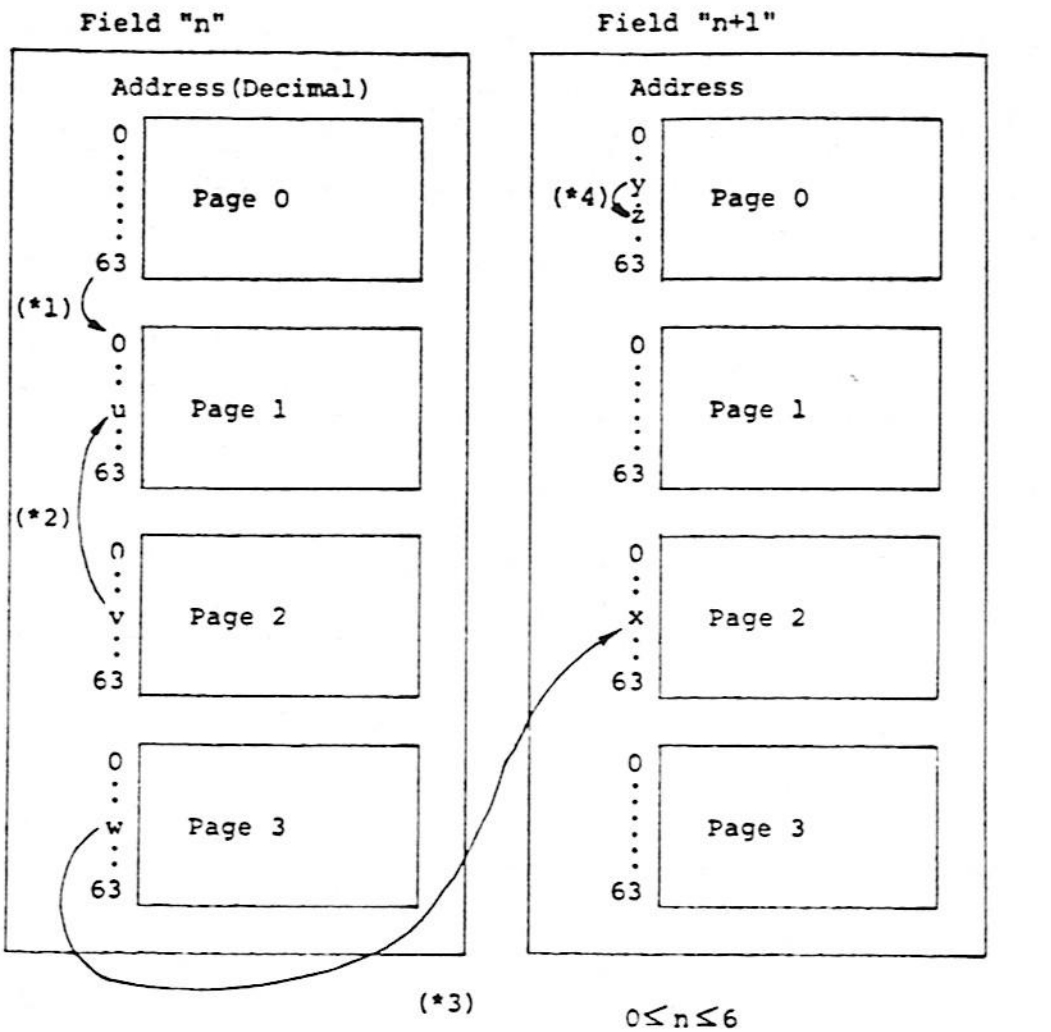
Upon system reset by the RESET input, the program counter is initialized to zero (ØØØ in Hex). Then, if the instruction is not a jump or a subroutine call instruction, the contents of the lower 8 bits of the program counter (8-bit binary up-counter) are simply incremented to execute the instructions sequentially. Since the 8-bit binary counter is automatically incremented and includes the 2 page-select bits and the 6 address-select bits, program flow automatically proceeds to address Ø of the next page after executing the instruction at the 64th address in the current page. Thus, in a field, a page is automatically extended to the next one and 4 pages (total of 256 words) are automatically executed. In order to extend the program flow into another field, the higher 3-bit register of the program counter must be modified with a JMP or CAL instruction. If not modified, the 8-bit binary counter is simply wrapped around to zero after executing the instruction at the 64th address in the 4th page, and the program counter then points to address Ø of page Ø in the same field. Thus, unless a transfer instruction is inserted, the 8-bit binary counter is simply incremented to execute instructions in order. In order to transfer the program flow to a different point, jump or subroutine call instructions are provided.

There are two types of jump instructions. The JMP instruction enables an unconditional jump to any address in the ROM area, rewriting all 11 bits of the program counter including the 3-bit field-select register. The JCP and JPA instructions enable jumps within a current page. The JCP instruction provides a jump to any one of 64 addresses in the current page. The JPA instruction provides a jump to one of 16 fixed addresses in the current page with the jump address being selected by the contents of the accumulator. The CAL instruction enables a subroutine call to any address in the ROM area, also rewriting all 11 bits of the program counter. The CZP instruction provides a subroutine call to one of sixty-four fixed addresses in field Ø, page Ø. In order to transfer to another page or another field, either the JMP or CAL instruction is used.

Hexadecimal and Binary Notation of Program Counter Contents.

| Field | Page | Hexadecimal Notation | Binary Notation | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Field | | | Page | | Address | | | | | | |
| | | | $PC_{10}$ | $PC_9$ | $PC_8$ | $PC_7$ | $PC_6$ | $PC_5$ | $PC_4$ | $PC_3$ | $PC_2$ | $PC_1$ | $PC_0$ |
| 0 | 0 | 0 0 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | 0 0 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | \| | | \| | | \| | | | \| | | | |
| | | 0 3 E | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| | | 0 3 F | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 1 | 0 4 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | 0 4 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | \| | | \| | | \| | | | \| | | | |
| | | 0 7 E | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| | | 0 7 F | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 2 | 0 8 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | 0 8 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | \| | | \| | | \| | | | \| | | | |
| | | 0 B E | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| | | 0 B F | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 3 | 0 C 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | 0 C 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | \| | | \| | | \| | | | \| | | | |
| | | 0 F E | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| | | 0 F F | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 — 3 | 1 0 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | \| | | \| | | \| | | | \| | | | |
| | | 1 F F | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 — 6 | | 2 0 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | \| | | \| | | \| | | | \| | | | |
| | | 6 F F | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 0 — 3 | 7 0 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | \| | | \| | | \| | | | \| | | | |
| | | 7 C F | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

## Program Transfer




*1) Program counter is automatically incremented and the program proceeds to address 0 of next page.

*2) Program is transferred to another page with the JMP instruction.

*3) Program is transferred to another field with the JMP instruction.

*4) Program is transferred within the same page with the JCP(or JMP) instruction.

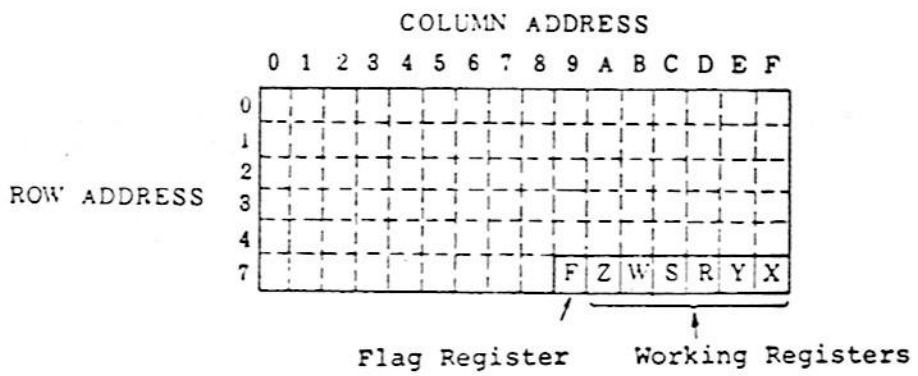
## DATA MEMORY (RAM)

The uCOM-43 contains 384 bits of static RAM for data storage. The RAM is organized as 96 words by 4 bits, and the 96 words are organized into 6 rows by 16 columns.

The RAM is addressed with the contents of the 7-bit data pointer. The higher 3 bits(DPH) of the data pointer specify the row address in order to select one of 6 rows. The lower 4 bits(DPL) of the data pointer specify the column address in order to select one of 16 columns.

Among the 96 words, a 4-bit word at address 79 (Hex) can be specifically used as a software controlled 4-bit flag register. Special instructions are provided that can directly set, reset and test any of the 4 flag bits. Another 6 words at addresses 7A to 7F can be specifically used as six 4-bit general purpose working registers. Various instructions are provided that can directly modify the specified working registers. The flag register and all 6 working registers may also be treated as normal RAM locations. All the uCOM-43 instructions that work with RAM data can commonly access all 96 locations, including the flag register and the working registers. In this case, all 96 RAM locations are addressed with the 7-bit data contained in the data pointer.

RAM Organization




The uCOM-43 instructions that work with RAM data provide the following functions. The arithmetic and logical operation instructions enable binary addition and logical exclusive OR between the accumulator and RAM data. The load and store

instructions enable data transfer and exchange between the accumulator and RAM locations. Two types of compare instructions are provided. The CMB instruction enables comparison of any single bit between the accumulator and RAM data. The program skips the next instruction if the compared two bits are equal. The CM instruction enables comparison of 4-bit data between the accumulator and a RAM location. The program skips the next instruction if both 4-bit data are equal.

There are two instructions that manipulate and test 4-bit RAM data. The INM instruction increments 4-bit data in a RAM location and the program skips if the 4-bit data is wrapped around to zero. The DEM instruction decrements 4-bit data in a RAM location and the program skips if the 4-bit data is decremented to F (Hex).

In order to provide efficient use of the memory space, bit set, reset and test capabilities are provided for any single bit in any RAM location. The SMB instruction sets and the RMB instruction resets a single bit in a desired RAM location. The TMB instruction tests any single bit in a desired RAM location and the next instruction is skipped if the tested bit is a one (1).

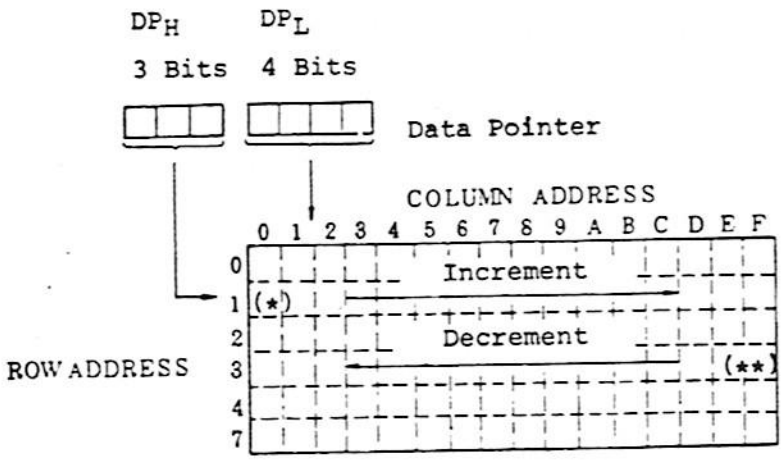
RAM MANIPULATION INSTRUCTIONS

| | | | | | |
|---|---|---|---|---|---|
| AD | LI | X | XI | INM | TMD |
| ADS | S | XM | XMI | DEM | |
| ADC | L | XD | CM | SMB | |
| EXL | LM | XMD | CMB | RMB | |

DATA POINTER (DP)

The 7-bit data pointer addresses one of 96 RAM (data memory) locations. The higher 3 bits (DPH) of the data pointer is a 3-bit register and its contents specify a row address to select one of 6 rows. The lower 4 bits (DPL) of the data pointer is a 4-bit up/down counter which specifies a column address to select one of 16 columns.

Data Pointer and RAM




*) Skip after increment.
**) Skip after decrement.

In order to provide powerful and flexible RAM addressing, a variety of instructions are provided that modify, increment, decrement, transfer, exchange, load immediate and test the contents of the data pointer.

The lower 2 bits of DPH can be logical exclusive-OR'ed with the operand (M1M0) of the LM, XM, XMD and XMI instructions.

Examples of DPH Modification.

1) Row address specified by DPH is modified from 1 to 3.

```
       DPH = 0 0 1  (Before modification)
(EXOR) M1M0=   1 0
-------------------
       DPH = 0 1 1  (After modification)
```

2) Row address specified by DPH is modified from 7 to 4.

```
       DPH = 1 1 1  (Before modification)
(EXOR) M1M0=   1 1
-------------------
       DPH = 1 0 0  (After modification)
```

The DPL is composed of a 4-bit up/down counter and can be incremented (+1), decremented (-1) and tested with instructions. The XI, XMI and IND instructions increment the contents of DPL and the next instruction is skipped if the contents of DPL are wrapped around to zero. The XD, XMD and DED instructions decrement the contents of DPL and the next instruction is skipped if the contents of DPL reach F (Hex).

Examples.

1) Increment DPL and skip using XI, XMI or IND instructions.

| | DPL | DPL |
|---|---|---|
| (Before increment) | 0 0 1 0 (2) | 1 1 1 1 (F) |
| (After increment) | 0 0 1 1 (3) | 0 0 0 0 (0) |
| | Non-skip | Skip next instruction. |

2) Decrement DPL and skip using XD, XMD or DED instructions.

| | DPL | DPL |
|---|---|---|
| (Before decrement) | 0 0 1 0 (2) | 0 0 0 0 (0) |
| (After decrement) | 0 0 0 1 (1) | 1 1 1 1 (F) |
| | Non-skip | Skip next instruction. |

These instructions' increment, decrement and test-and-skip functions for the DPL will enable efficient programming of counting loops such as those used in digit counting for arithmetic operations. The contents in the data pointer and the working registers can be exchanged by the working register instructions. The XHX and XHR instructions exchange the contents of DPH and the X and R registers, respectively. The XLY and XLS instructions exchange the contents of DPL and the Y and S registers, respectively. With these instructions, the data pointer contents can be saved upon interrupt acknowledge and the working registers contents can be stored in the data pointer if necessary during arithmetic operations.

Direct and indirect RAM addressing is also made available with data pointer manipulation instructions. The LDI instruction loads the data pointer with 7 bits of immediate data, and the LDZ instruction loads DPH with 0 and DPL with 4 bits of immediate data for direct RAM addressing. The TAL and TLA instructions enable 4-bit data transfer between the DPL and the accumulator. These instructions provide indirect RAM addressing using the contents of accumulator.

With the compare instruction CLI, the contents in DPL and 4-bits of immediate data can be compared and tested. If both are equal, the next instruction is skipped.

The contents in DPL are also used to select one of 9 input/output ports in input/output instructions such as SPB, RPB, TPB, OP and IP.

USES OF THE DPL REGISTER

Counting Loops (XI, XMI, IND, XD, XMD, DED)

Interrupt Saving of DPL (XLY, XLS)

Saving of Working Register (XLY, XLS)

Direct RAM Addressing (LDI, LDZ)

Indirect RAM Addressing (TAL, TLA)

Immediate Comparison (CLI)

I/O Selection (SPB, RPB, TPB, OP, IP)

## STACK REGISTER

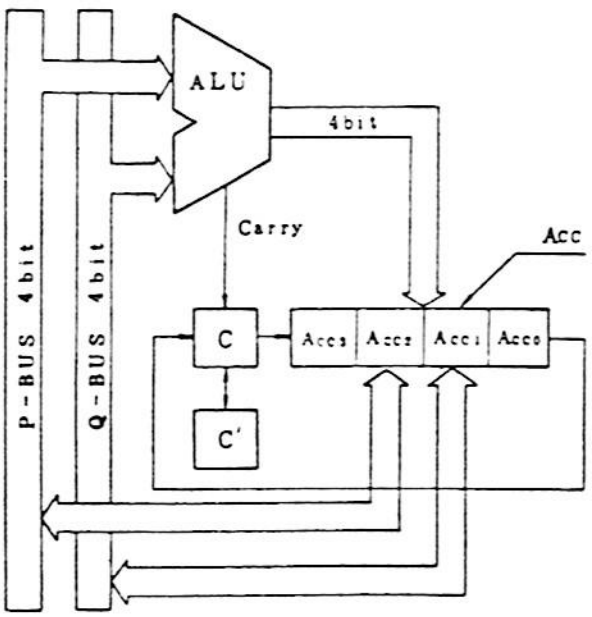
The stack register is a last-in-first-out (LIFO) push down stack register organized as 3 words by 11 bits. This register is used to save the contents of the 11-bit program counter (return address) when a subroutine is called or an interrupt is acknowledged. The stack register enables 3 levels of stacking of return addresses. All 3 levels may be used for nesting of subroutines unless the system is interrupt driven. Then a level in the stack register may be used to save the return address for an interrupt service and the other 2 levels may be used for subroutine nesting. If more than 3 levels of return addresses are nested, the first-in address is lost.

The contents of the program counter is pushed onto the stack when a subroutine is called with the CZP or CAL instruction and when an interrupt is acknowledged. The stacked return address is restored to the program counter when the program is returned with an RT or RTS instruction.

## ALU AND ACCUMULATOR (ACC)

The ALU (arithmetic logic unit) and the accumulator (ACC) form the heart of the uCOM-43 microcomputer system. The ALU performs arithmetic and logical operations and tests for operation results. The result of an operation by the ALU is stored in the ACC and in the carry F/F. The ACC is a 4-bit register which stores ALU results and other data to be processed. The carry F/F is a 1-bit flip-flop which indicates when a carry bit is generated during addition.

**ALU and Accumulator**




The ALU can perform the following functions as specified by appropriate instructions.

Binary addition.

Increment (+1) and decrement (-1).

Logical exclusive OR.

Decimal adjustment for addition (+6) and subtraction (+10).

Comparison.

Bit test.

Complement.

Three instructions are provided for binary addition. The AD instruction adds the memory data pointed to by the data pointer to the Acc in binary. The ADS and ADC instructions add the memory data pointed to by the data pointer to the Acc together with the carry bit in binary. The carry F/F is affected by the results of the ADS and ADC instructions. (Among the arithmetic and logical operation instructions, only ADS and ADC instructions affect the carry flag.)

Binary subtraction can be done by performing one's or two's complement addition with the use of the CMA or CIA instructions, respectively. Decimal addition and subtraction can be performed using the decimal adjust instructions, DAA (+6) for addition and DAS (+10) for subtraction, respectively. The EXL instruction performs logical exclusive OR between the Acc contents and memory data pointed to by the data pointer.

The Acc can be incremented (+1) and decremented (-1). The INC instruction increments the contents of the Acc and the next instruction is skipped if the Acc is wrapped around to 0. The DEC instruction decrements the contents of the Acc and the next instruction is skipped if the Acc reaches F (Hex). With the RAR instruction, the 4-bits in the Acc can be rotated one bit right through the carry bit.

The load and store instructions enable data transfer and exchange between the Acc and the RAM location addressed by the data pointer. The LI instruction loads the Acc with 4-bits of immediate data. The data pointer manipulation instructions TAL and TLA provide a 4-bit data transfer between the Acc and DPL. Also, the working register manipulation instructions provide for data transfer and exchange between the Acc and the working registers Z and W.

The bit manipulation instruction TAB enables conditional skip by testing any single bit in the Acc. The CMB instruction also provides conditional skip by comparing any single bit of the Acc with the corresponding bit in the RAM data pointed to by the data pointer. Other conditional skip operations are also made possible with compare instructions. With the CM instruction, the program skips if the 4 bits in the Acc and a RAM location pointed to by the data pointer are equal. The CI instruction provides a conditional skip by comparing the data in the Acc and an immediate 4-bit data field.

The Acc is also used as the source and destination for data transfer with the external world. With the use of the input/output instructions, data can be loaded into the Acc via input ports and the data generated or processed in the processor can be sent out from the Acc via output ports.

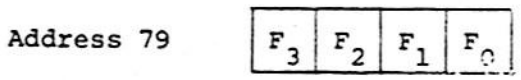
A 1-bit carry save F/F(C') is provided in addition to the carry F/F(C) which is used in normal program operations. The carry save F/F(C') is used to save the latest status of the carry

F/F(C) upon a subroutine call or an interrupt acknowledge. The XC instruction exchanges the contents of C and C'. The TC instruction provides for a conditional skip on the carry F/F by testing the carry flag and causing a skip if it is set (1).

## FLAG REGISTER

A 4-bit word located at address 79 (Hex) in the RAM can be specifically used as a software controlled general purpose flag register. Four types of flag manipulation instructions are provided (SFB, RFB, FBT, FBF). These can operate directly upon any single bit in the flag register without loading the data pointer with address 79.

**Flag Register**



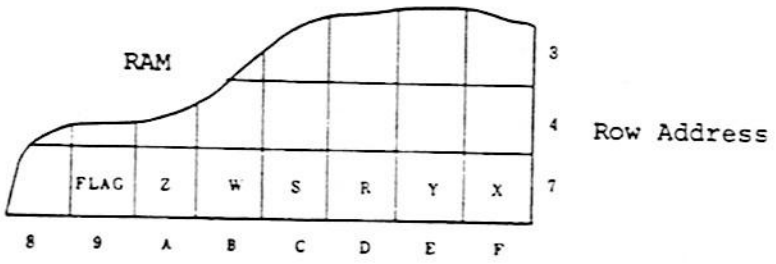

The SFB instruction sets and the RFB instruction resets a desired single bit in the flag register. The FBT and FBF instructions test a single flag bit and cause a skip if the tested bit is true (1) or false (0), respectively.

If more than 4 flag bits are necessary, other RAM locations may be used as software controlled flag registers. Bit manipulation instructions are provided to set, reset and test a desired single bit in any RAM word addressed by the data pointer.

## WORKING REGISTERS (WR)

The 6 words located at addresses 7A to 7F (Hex) in RAM can be used as six 4 bit general purpose working registers, Z, W, S, R, Y and X, respectively. With the working register manipulation instructions, these working registers are directly addressed independent of the data pointer.

**Location of Working Registers in the RAM**




**Column Address**

The working register manipulation instructions enable data transfer and exchange between the data pointer and the working registers X, Y, R and S, and also between the ACC and the working registers Z and W. With these instructions, the working registers can be used to save the latest contents of the data pointer and the ACC when an interrupt is acknowledged. The working registers can also be used as temporary storage for the ACC contents in order to ease programming in various occasions.

WORKING REGISTER INSTRUCTIONS

| | | | | |
|---|---|---|---|---|
| XHX | THX | XAZ | TAZ | XHR |
| XLY | TLY | XAW | TAW | XLS |

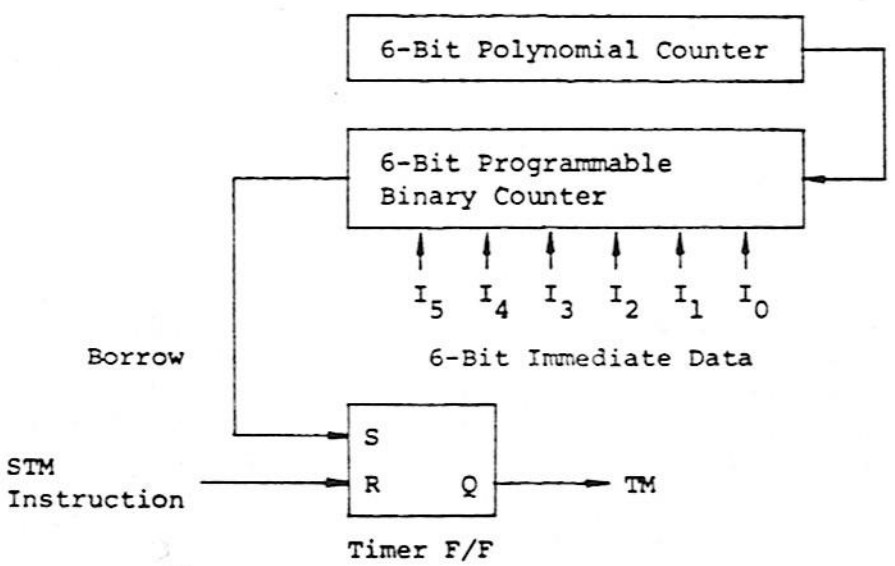
## PROGRAMMABLE INTERVAL TIMER

The uCOM-43 contains a software programmable interval timer composed of a 6-bit polynomial counter and a 6-bit programmable binary counter.

The initial setting of the timer is done using the timer set instruction STM, with the timer starting to count at the end of the STM instruction execution. The STM instruction contains 6 binary bits of immediate data (I5-IØ) which is loaded in the 6-bit programmable binary counter upon STM instruction execution. By varying the 6-bit immediate data, one of 64 time intervals can be programmed.

If the clock frequency for the uCOM-43 is set at 4ØØKHz, the minimum time interval is 63Øusec and the maximum interval is 4Ø.32msec with increments of 63Øusec. If a longer time interval is necessary than available with the built-in interval timer,

**Interval Timer Configuration**

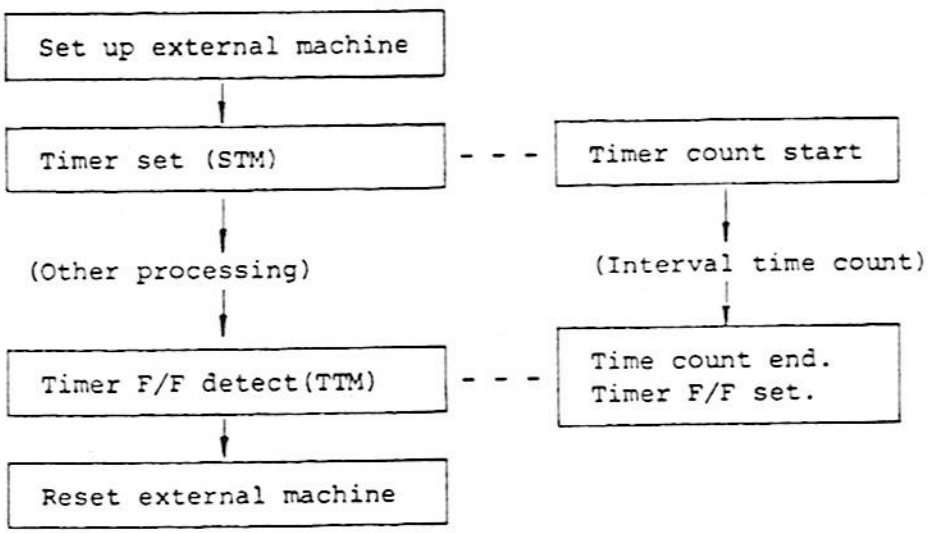


it can be obtained by using a portion of the RAM area as a counter. Then the built-in timer would be used as the basic timer, incrementing RAM each time it executed the basic count.

The built-in interval timer has a timer F/F(TM F/F). The TM F/F is reset by the STM instruction and is set when the time interval specified by the STM instruction has passed. To detect the condition of the TM F/F and perform the necessary servicing when the time has passed, the timer test instruction TTM is provided. The TTM instruction tests the TM F/F and causes a skip if it is set.

Thus, by use of the interval timer and the STM and TTM instructions, program operations not related to timing can be performed in parallel. The built-in timer eliminates the need for an external timer circuit or a software timer, and it can be set up and detected with very simple programming steps. By the efficient use of these capabilities, the throughput of the uCOM-43 system can be greatly increased. The interval timer can be used to control external tasks that require timing control. To set up an external machine and reset it after a desired interval time, for a very simple example, the following procedure may be used.

Control of External Machine with the Timer




The details of the interval timer are as follows: Upon STM instruction execution, the 6-bit polynomial counter and the TM F/F are cleared. The 6-bit immediate data (I5-I0) of the STM instruction is loaded into the 6-bit programmable binary counter. The interval timer is thus initialized by the STM instruction and starts counting at the end of the STM instruction execution. The 6-bit polynomial counter counts up by one at the end of every machine cycle. One machine cycle equals 10usec if the clock frequency is set at 400KHz. Every time the polynomial counter

reaches 63 in decimal, it sends a signal to the binary counter and is cleared to continue counting to 63 again. Thus, counting from 1 to 63 operates the basic time interval (63 x 10usec = 630usec at 400KHz clock).

The 6-bit binary counter is decremented by one whenever the signal is sent from the polynomial counter indicating its counting up to 63. When a borrow is generated in the binary counter, counting ceases and the TM F/F is set. If the binary counter is set to zero (000000) with the STM instruction, it is decremented by the first signal from the polynomial counter and a borrow is generated. In this case, the timer count is ended with the minimum programmable time interval (630usec) and the TM F/F is set for detection by the TTM instruction. If the STM instruction programs the binary counter to 000001, 63 x 10usec x 2 = 1,260 usec are available. If the maximum value 111111 is programmed, 63 x 10usec x 64 = 40,320usec are counted.

The TM F/F is reset only by the STM instruction. Thus, unless the interval timer is reinitialized with the STM instruction, the TM F/F remains in a set state. The interval timer can be restarted even while the previous time count is continued (thus the TM F/F is in a reset state) by specifying a new STM instruction. The immediate data of new STM instruction overrides the time count in the binary counter and the timer is restarted for the newly programmed time interval. Thus the previous timing is aborted.

**Immediate Data of STM Instruction and Interval Time**

(When clock frequency is 400 KHz.)

| Decimal I | Immediate Data (Binary) $I_5$ | $I_4$ | $I_3$ | $I_2$ | $I_1$ | $I_0$ | Programmed Interval Time (μsec) |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 630 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1,260 |
| 2 | 0 | 0 | 0 | 0 | 1 | 0 | 1,890 |
| ⋮ | | | | | | | ⋮ |
| 62 | 1 | 1 | 1 | 1 | 1 | 0 | 39,690 |
| 63 | 1 | 1 | 1 | 1 | 1 | 1 | 40,320 |

Calculation of programmed interval time.

I = Decimal notation of 6-bit immediate data.

1) When clock frequency = 400 KHz ;

Programmed interval time = 0.63(I + 1) msec.

2) When clock frequency = f KHz ;

Programmed interval time = $252 \frac{I + 1}{f}$ msec.

## INPUT/OUTPUT PORTS

The uCOM-43 has 35 input/output channels for communication with and control of the external world. These ports are organized into 9 input/output ports (A to I). Eight ports (A to H) are composed of 4 bits each and the last port (I) is composed of 3 bits.

| | | |
|---|---|---|
| Input Ports | (4 Bits Each) | : A, B |
| Input/Output Ports | (4 Bits Each) | : C, D |
| Output Ports | (4 Bits Each) | : E, F, G, H |
| Output Ports | (3 Bits) | : I |

In order to provide flexible and efficient use of these I/O ports, a variety of input/output instructions are provided which enable single bit set/reset, single bit test and conditional skip, 4-bit parallel input/output and 8-bit immediate parallel output. The I/O instructions are divided into two types; the ones dedicated to specific ports and the ones that use the 4-bit data in the DPL to select a desired port. The former include such instructions as IA and OE that specifically access port A and E, respectively. The latter require that a 4-bit code assigned to the desired port be loaded into the DPL using data pointer manipulation instructions prior to I/O instruction execution.

All of the I/O instructions that work on a single bit contain 2-bit immediate data (B1B0) to specify a single bit in the accessed port.

I/O instructions dedicated to specific ports are:

| | | |
|---|---|---|
| SEB, REB, OE | : | Output Port E. |
| OCD | : | Output Port C and D. |
| TPA, IA | : | Input Port A. |

I/O instructions that use DPL to select a port are:

SPB, RPB, TPB, OP and IP.

Correspondence of port and 4-bit code in DPL

| Port | Code in DPL Binary | Hex |
|---|---|---|
| A | 0000 | 0 |
| B | 0001 | 1 |
| C | 0010 | 2 |
| D | 0011 | 3 |
| E | 0100 | 4 |
| F | 0101 | 5 |
| G | 0110 | 6 |
| H | 0111 | 7 |
| I | 1000 | 8 |

The function of each I/O port is summarized below. The instruction mnemonics for each operation are in parentheses.

Input Ports A and B.

1) 4-bit parallel input from the specified port to the Acc. (IA and IP)
2) Test any single bit and skip if it is "1". (TPA and TPB)
3) Port A is directly accessed without use of the DPL. (IA and TPA)

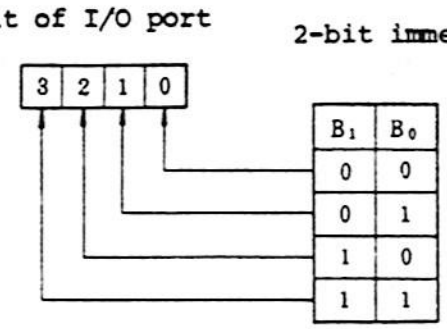
Input/Output Ports C and D.

1) 4-bit parallel input from the specified port to the Acc. (IP)
2) Test any single bit and skip if it is "1". (TPB)
3) 4-bit parallel output from the Acc to the specified port. (OP) The output data on the port is latched and held valid until the next I/O instruction to that port.
4) The 8-bit immediate data (I7-I0) of the instruction is output to port D(I7-I4) and C(I3-I0) in parallel. (OCD)
5) Switching of the port to either input or output mode is done automatically by the I/O instruction.

Output Ports E, F, G, H and I.

1) 4-bit parallel output from the Acc to the specified port. (OP and OE) Port I outputs the lower 3 bits of the Acc. The output data on the port is latched and held valid until the next output instruction to that port.
2) Set and reset any desired single bit. (SPB, RPB, SEB and REB)
3) Port E can be directly accessed without use of the DPL. (OE, SEB and REB)

Correspondence of selected bit and 2-bit immediate data in the I/O instructions.




I/O Instructions and Functions of I/O Ports. ( O = Available Function )

| I/O Instruction | In/Out | I | I | I/O | I/O | O | O | O | O | O** |
|---|---|---|---|---|---|---|---|---|---|---|
| | Port | A | B | C | D | E | F | G | H | I** |
| | $DP_L$*** Function | 0000 0 | 0001 1 | 0010 2 | 0011 3 | 0100 4 | 0101 5 | 0110 6 | 0111 7 | 1000 8 |
| SEB | Single Bit Set | | | | | O* | | | | |
| REB | Single Bit Reset | | | | | O* | | | | |
| SPB | Single Bit Set | | | | | O | O | O | O | O |
| RPB | Single Bit Reset | | | | | O | O | O | O | O |
| TPA | Single Bit Test | O* | | | | | | | | |
| TPB | Single Bit Test | O | O | O | O | | | | | |
| OE | 4-Bit Parallel Output | | | | | O* | | | | |
| OP | 4-Bit Parallel Output | | | O | O | O | O | O | O | O** |
| OCD | 8-Bit Immediate Output | | | O* | O* | | | | | |
| IA | 4-Bit Parallel Input | O* | | | | | | | | |
| IP | 4-Bit Parallel Input | O | O | O | O | | | | | |

Note : *) The port is directly accessed without use of $DP_L$.

**) The port I consists of 3 bits. Others are all 4 bits organization.

***) The port is selected with the 4 bit code in $DP_L$. Shown in the table are binary and hexadecimal notation of the code.
In single bit manipulation, bit selection is done with the 2-bit immediate data($B_1B_0$) of the instruction.

SYSTEM CONTROL PINS

The uCOM-43 has four system control pins, the functions and use of which are described in this section.

Clock Control Inputs (CL1 and CLØ)

Reset Input (RES)

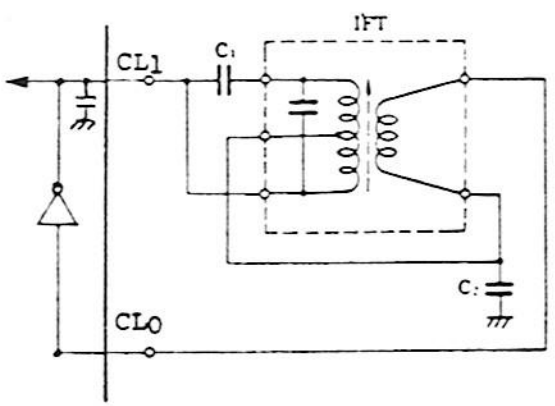
Interrupt Request Input (INT-NOT)

CLOCK CONTROL INPUTS (CL1 and CLØ)

When the uCOM-43 is used in systems that do not require precise timing, the internal clock generation circuit of the uCOM-43 can easily be driven by connecting a simple LC circuit to the CL1 and CLØ inputs.

If an LC circuit is to be used, an economical IFT (intermediate frequency transformer) and capacitors may be connected to form a Hartley oscillation circuit as illustrated. The capacitance on the CL1 input line is contained on the chip. The less noise on the CLØ input, the more stable the oscillation will be. The oscillation frequency generated by the oscillator equals the clock frequency of the uCOM-43, and one machine cycle is defined as four clock cycles.

Hartley Oscillator

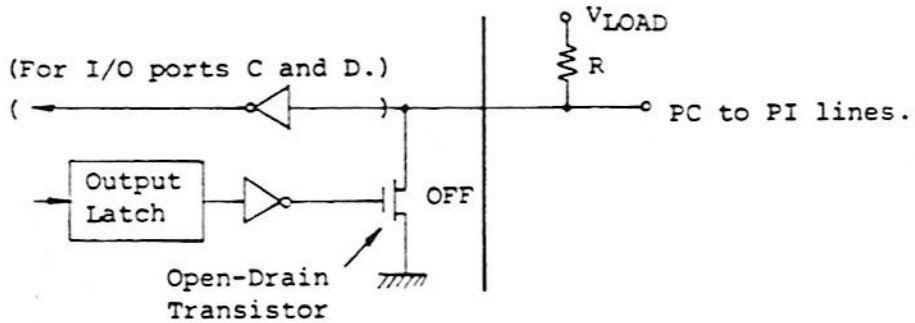



If the uCOM-43 is to be used in systems requiring precise timing, an external MOS level clock signal may be provided to the CLØ input. In this case the CL1 line should be left open. One machine cycle again equals four clock cycles.

RESET INPUT (RES)

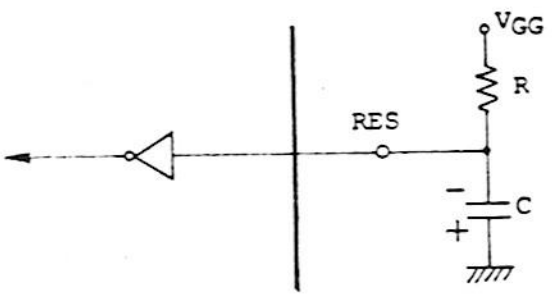
The uCOM-43 system is reset by providing a "High" level input equal to or longer than 4 machine cycles (16 clock cycles) to the RES input. Upon reset, the program counter (PC) is loaded with address 000 (field 0, page 0 and address 0). Also, the INT F/F and INTE F/F are both reset to inhibit interrupts. The open-drain transistors of all 27 output lines (Ports C to I) are turned "OFF" as illustrated.

Open-Drain Transistors of Output Port C to I upon Reset




When the RES input is removed, the program starts its operation from address 000. With the simple external RC circuit shown, automatic power-on-reset is possible.

Power-On-Reset Circuit

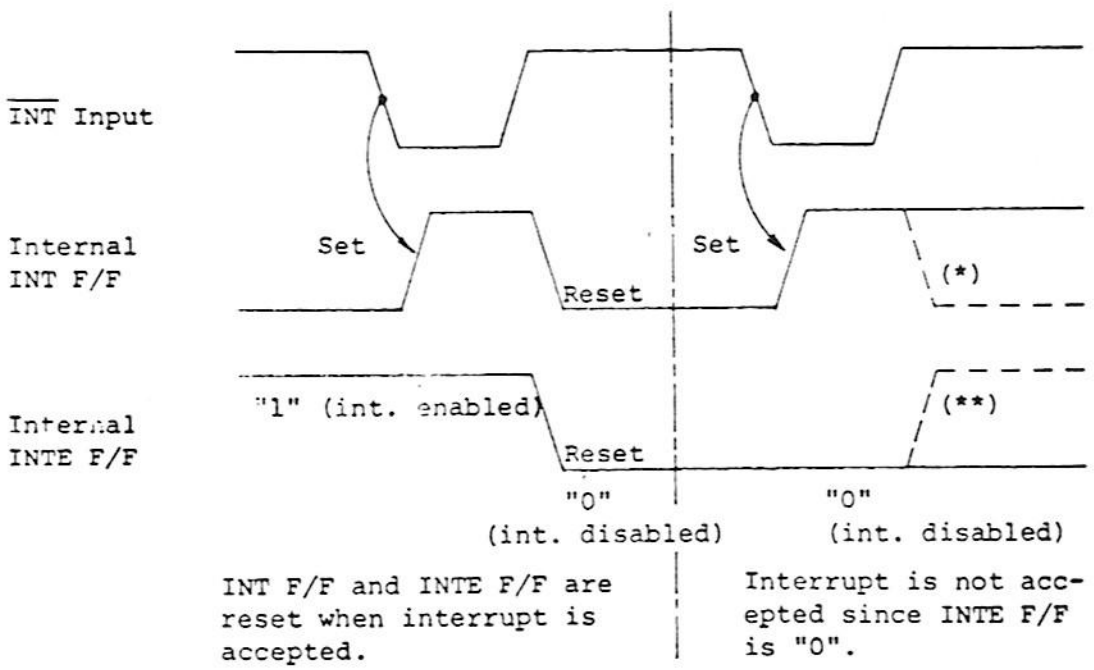


## INTERRUPT REQUEST INPUT (INT-NOT)

In order to enable efficient interrupt control, the uCOM-43 has two flip-flops, INT F/F (interrupt F/F) and INTE F/F (interrupt enable F/F). These are controlled and tested by the three interrupt handling instructions, EI (enable interrupt), DI (disable interrupt) and TIT (test interrupt F/F).

The INT F/F is set whenever an interrupt request (a going low transition) is applied to the INT input line, and is reset when an interrupt is accepted or a TIT instruction is executed. The INTE F/F is fully software controlled. It is set by the EI instruction and is reset by the DI instruction or upon interrupt acceptance. While the INTE F/F is reset, it masks the condition of the INT F/F and disables interrupts. The INT F/F and INTE F/F are set and reset independently and have no affect on each other. The INT F/F is set whenever an interrupt is requested. The condition of INT F/F is sampled during the last clock cycle of every instruction. If the INT F/F is a "1" and interrupts are enabled (INTE F/F = "1") at this time, the program jumps to the interrupt service routine after completing the current instruction. If interrupts are disabled (INTE F/F = "Ø"), the condition of the INT F/F is neglected and the interrupt is not accepted.

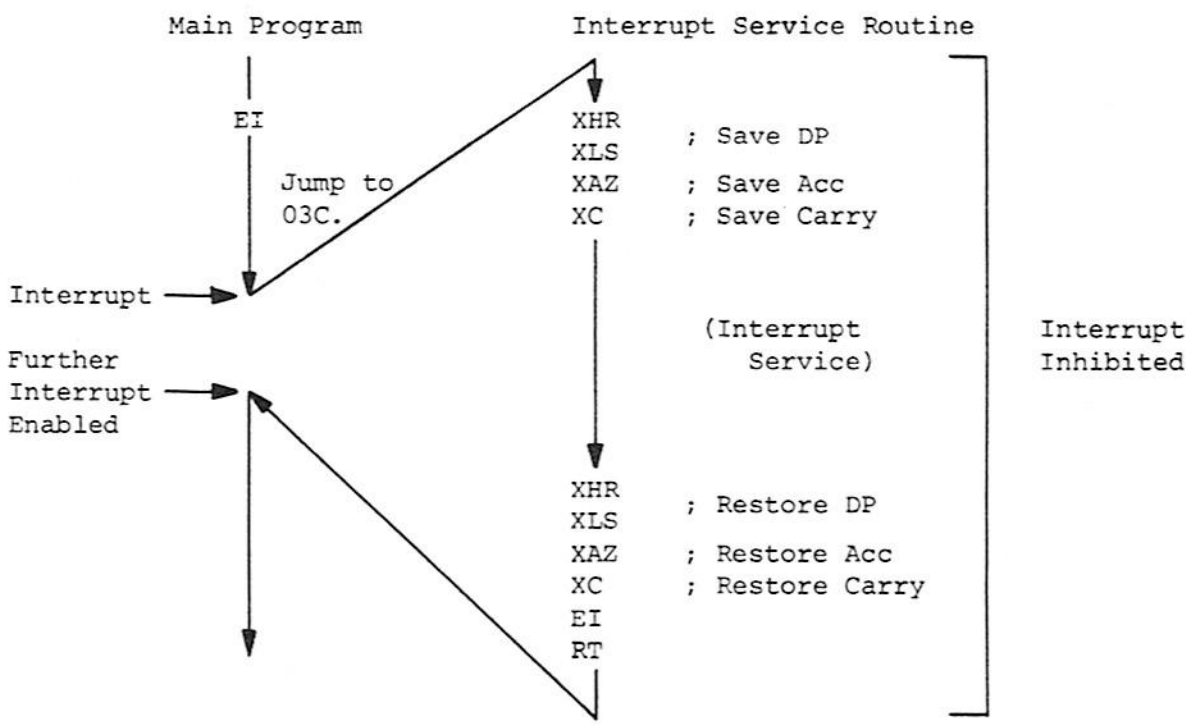
INTERRUPT TIMING




Note : *) INT F/F is reset by TIT instruction.
**) INTE F/F is set by EI instruction and is reset by DI instruction.

When an interrupt is accepted, the contents of the program counter (the address for the next instruction) is saved in the push-down stack register and the fixed address 03C (field 0, page 0, address 60) is loaded into the program counter. The INT F/F and INTE F/F are both reset to inhibit further interrupt acceptance. The program then jumps to address 03C where the first instruction of the interrupt service routine should be located. At the beginning and end of the interrupt service routine, the exchange instructions may be inserted to save and restore the data pointer, Acc and carry flag contents of the main program routine. Also, an EI instruction may be inserted in the interrupt service routine just before returning to the main routine with the RT (return) instruction. The EI instruction enables interrupt acknowledgement only after completing the RT instruction. The RT instruction restores the last saved program return address from the stack register to the program counter.

Interrupt Service Routine Example



The condition of the INT F/F is sampled during the last clock cycle of every instruction execution cycle (except two instructions CLA and LI). In the case of multiple machine cycle instructions including conditional skip instructions, the INT F/F is sampled at the end of the last machine cycle. If an interrupt is pending (INT F/F = "1") and interrupt is enabled (INTE F/F = "1") when sampled, the interrupt is acknowledged. The only exceptions are the CLA and LI instructions. During all single and successive CLA and LI instructions, the condition of INT F/F is neglected independent of the status of INTE F/F and the interrupt is not accepted.

The TIT instruction is provided in order to enable an interrupt to be serviced by software even while interrupts are disabled (INTE F/F = "Ø"). This instruction may be used to implement a conditional interrupt service or to honor an interrupt only in a specific program area. The TIT instruction tests the INT F/F, resets it at T3 of the first machine cycle, and causes a skip over the next instruction if it has been set.

Note: The INT F/F is reset by the TIT instruction but may be set again during the TIT instruction by another interrupt request. The condition of INT F/F is automatically sampled at the last cycle (T4) of the TIT instruction and the new interrupt is accepted.

In order to avoid overstacking, the interrupt request is not acknowledged during the EI instruction. The INTE F/F is set at the start of the instruction following EI, and sampling of the INT F/F condition starts during the last cycle of the instruction following the EI. Thus, if the RT instruction follows EI at the end of an interrupt service routine, the next interrupt is accepted only after completing the RT instruction.

INSTRUCTION SET SUMMARY

**Note :**

| | |
|---|---|
| **Acc** | **: Accumulator** |
| **An** | **: Accumulator bit n** |
| **C** | **: Carry F/F** |
| **C'** | **: Carry save F/F** |
| **Carry** | **: Carry bit generated in Acc, (but does not necessarily set C).** |
| **Borrow** | **: Borrow generated in Acc, (but does not affect on C).** |
| **DP** | **: Data pointer** |
| **PC** | **: Program counter** |
| **X** | **: Working register X** |
| **Y** | **: Working register Y** |
| **Z** | **: Working register Z** |
| **W** | **: Working register W** |
| **R** | **: Working register R** |
| **S** | **: Working register S** |
| **FLAG** | **: Flag register** |
| **TM F/F** | **: Timer F/F** |
| **TIMER** | **: Timer** |
| **INT F/F** | **: Interrupt F/F** |
| **INTE F/F** | **: Interrupt Enable F/F** |
| **STACK** | **: Stack register** |
| **PORT** | **: Input/output port** |
| **( )** | **: Contents** |
| **[(XX)]** | **: Memory data addressed by (DP) or the bit of the port specified by (DP, $B_1B_0$).** |
| ← | **: Transfer or transfer of result.** |
| ⇄ | **: Exchange** |
| ⊻ | **: Logical exclusive OR.** |
| **1** | **: Set** |
| **0** | **: Reset** |

Mnemonic. Instruction Code. Bytes. Machine Cycles. Operation. Condition for Skip.

| Mnemonic | $D_7D_6D_5D_4$ | $D_3D_2D_1D_0$ | Bytes | Machine Cycles | Operation | Condition for Skip |
|---|---|---|---|---|---|---|
| CLA | 1 0 0 1 | 0 0 0 0 | 1 | 1 | Acc←0 | |
| CLC | 0 0 0 0 | 1 0 1 1 | 1 | 1 | C←0 | |
| CMA | 0 0 0 1 | 0 0 0 0 | 1 | 1 | Acc←($\overline{\text{Acc}}$) | |
| CIA | 0 0 0 1 | 0 0 0 1 | 1 | 1 | Acc←($\overline{\text{Acc}}$)+1 | |
| INC | 0 0 0 0 | 1 1 0 1 | 1 | 1/2~3 | Acc←(Acc)+1<br>skip if Carry | Carry |
| DEC | 0 0 0 0 | 1 1 1 1 | 1 | 1/2~3 | Acc←(Acc)−1<br>skip if Borrow | Borrow |
| STC | 0 0 0 1 | 1 0 1 1 | 1 | 1 | C←1 | |
| XC | 0 0 0 1 | 1 0 1 0 | 1 | 1 | (C)←(C′) | |
| RAR | 0 0 1 1 | 0 0 0 0 | 1 | 1 | (C)→$Acc_3$, $(Acc_n)\rightarrow(Acc_{n-1})$, $(Acc_0)\rightarrow C$ | |
| INM | 0 0 0 1 | 1 1 0 1 | 1 | 1/2~3 | [(DP)]←[(DP)]+1<br>skip if [(DP)]=0 | [(DP)]=0 |
| DEM | 0 0 0 1 | 1 1 1 1 | 1 | 1/2~3 | [(DP)]←[(DP)]−1<br>skip if [(DP)]=F | [(DP)]=F |
| AD | 0 0 0 0 | 1 0 0 0 | 1 | 1/2~3 | Acc←(Acc)+[(DP)]<br>skip if Carry | Carry |
| ADS | 0 0 0 0 | 1 0 0 1 | 1 | 1/2~3 | Acc, C←(Acc)+[(DP)]+(C)<br>skip if Carry | Carry |
| ADC | 0 0 0 1 | 1 0 0 1 | 1 | 1 | Acc, C←(Acc)+[(DP)]+(C) | |
| DAA | 0 0 0 0 | 0 1 1 0 | 1 | 1 | Acc←(Acc)+6 | |
| DAS | 0 0 0 0 | 1 0 1 0 | 1 | 1 | Acc←(Acc)+10 | |
| EXL | 0 0 0 1 | 1 0 0 0 | 1 | 1 | Acc←(Acc)∀[(DP)] | |
| LI | 1 0 0 1 | $I_3I_2I_1I_0$ | 1 | 1 | Acc←$I_3I_2I_1I_0$ | |
| S | 0 0 0 0 | 0 0 1 0 | 1 | 1 | [(DP)]←(Acc) | |
| L | 0 0 1 1 | 1 0 0 0 | 1 | 1 | Acc←[(DP)] | |
| LM | 0 0 1 1 | 1 0 $M_1M_0$ | 1 | 1 | Acc←[(DP)]<br>$DP_H$←$(DP_H)\forall M_1M_0$ | |

Note : When condition for skip is met, the skip instruction requires additional 1 to 2 machine cycles depending on the instruction to be skipped.

Mnemonic. Instruction Code. Bytes. Machine Cycles. Operation. Condition for Skip.

| | | $D_7D_6D_5D_4$ | $D_3D_2D_1D_0$ | | | | |
|---|---|---|---|---|---|---|---|
| | X | 0 0 1 0 | 1 0 0 0 | 1 | 1 | (Acc)←((DP)) | |
| | XM | 0 0 1 0 | 1 0 $M_1M_0$ | 1 | 1 | (Acc)←((DP))<br>$DP_H$←$(DP_H)\forall M_1M_0$ | |
| | XD | 0 0 1 0 | 1 1 0 0 | 1 | 1½~3 | (Acc)←((DP))<br>$DP_L$←$(DP_L)-1$<br>skip if $(DP_L)=F$ | $(DP_L)=F$ |
| | XMD | 0 0 1 0 | 1 1 $M_1M_0$ | 1 | 1½~3 | (Acc)←((DP))<br>$DP_H$←$(DP_H)\forall M_1M_0$<br>$DP_L$←$(DP_L)-1$<br>skip if $(DP_L)=F$ | $(DP_L)=F$ |
| | XI | 0 0 1 1 | 1 1 0 0 | 1 | 1½~3 | (Acc)←((DP))<br>$DP_L$←$(DP_L)+1$<br>skip if $(DP_L)=0$ | $(DP_L)=0$ |
| | XMI | 0 0 1 1 | 1 1 $M_1M_0$ | 1 | 1½~3 | (Acc)←((DP))<br>$DP_H$←$(DP_H)\forall M_1M_0$<br>$DP_L$←$(DP_L)+1$<br>skip if $(DP_L)=0$ | $(DP_L)=0$ |
| * | LDI | 0 0 0 1<br>0 $I_6$ $I_5$ $I_4$ | 0 1 0 1<br>$I_3I_2I_1I_0$ | 2 | 2 | DP←$I_6$~$I_0$ | |
| | LDZ | 1 0 0 0 | $I_3I_2I_1I_0$ | 1 | 1 | $DP_H$←0<br>$DP_L$←$I_3I_2I_1I_0$ | |
| | DED | 0 0 0 1 | 0 0 1 1 | 1 | 1½~3 | $DP_L$←$(DP_L)-1$<br>skip if $(DP_L)=F$ | $(DP_L)=F$ |
| | IND | 0 0 1 1 | 0 0 1 1 | 1 | 1½~3 | $DP_L$←$(DP_L)+1$<br>skip if $(DP_L)=0$ | $(DP_L)=0$ |
| | TAL | 0 0 0 0 | 0 1 1 1 | 1 | 1 | $DP_L$←(Acc) | |
| | TLA | 0 0 0 1 | 0 0 1 0 | 1 | 1 | Acc←$(DP_L)$ | |
| | XHX | 0 1 0 0 | 1 1 1 1 | 1 | 2 | $(DP_H)$=(X) | |
| | XLY | 0 1 0 0 | 1 1 1 0 | 1 | 2 | $(DP_L)$=(Y) | |
| | THX | 0 1 0 0 | 0 1 1 1 | 1 | 2 | $(DP_H)$→X | |
| | TLY | 0 1 0 0 | 0 1 1 0 | 1 | 2 | $(DP_L)$→Y | |
| | XAZ | 0 1 0 0 | 1 0 1 0 | 1 | 2 | (Acc)=(Z) | |
| | XAW | 0 1 0 0 | 1 0 1 1 | 1 | 2 | (Acc)=(W) | |

* When more than one LDI instruction occurs in sequence, only the first instruction encountered will be executed. The remaining LDI instructions in the string will be executed as NOPs.

Mnemonic. Instruction Code. Bytes. Machine Cycles. Operation. Condition for Skip.

| | $D_7D_6D_5D_4$ | $D_3D_2D_1D_0$ | | | | |
|---|---|---|---|---|---|---|
| TAZ | 0 1 0 0 | 0 0 1 0 | 1 | 2 | (Acc)→Z | |
| TAW | 0 1 0 0 | 0 0 1 1 | 1 | 2 | (Acc)→W | |
| XHR | 0 1 0 0 | 1 1 0 1 | 1 | 2 | $(DP_H) \rightleftarrows (R)$ | |
| XLS | 0 1 0 0 | 1 1 0 0 | 1 | 2 | $(DP_L) \rightleftarrows (S)$ | |
| SMB | 0 1 1 1 | 1 0 $B_1B_0$ | 1 | 1 | {(DP, $B_1B_0$)}←1 | |
| RMB | 0 1 1 0 | 1 0 $B_1B_0$ | 1 | 1 | {(DP, $B_1B_0$)}←0 | |
| TMB | 0 1 0 1 | 1 0 $B_1B_0$ | 1 | 1½~4 | skip if {(DP, $B_1B_0$)}=1 | {(DP, $B_1B_0$)} = 1 |
| TAB | 0 0 1 0 | 0 1 $B_1B_0$ | 1 | 1½~4 | skip if (Acc($B_1B_0$))=1 | (Acc($B_1B_0$)) = 1 |
| CMB | 0 0 1 1 | 0 1 $B_1B_0$ | 1 | 1½~4 | skip if (Acc($B_1B_0$)) ≠ {(DP, $B_1B_0$)} | (Acc($B_1B_0$)) ≠ {(DP, $B_1B_0$)} |
| SFB | 0 1 1 1 | 1 1 $B_1B_0$ | 1 | 2 | FLAG($B_1B_0$)←1 | |
| RFB | 0 1 1 0 | 1 1 $B_1B_0$ | 1 | 2 | FLAG($B_1B_0$)←0 | |
| FBT | 0 1 0 1 | 1 1 $B_1B_0$ | 1 | 2½~4 | skip if (FLAG($B_1B_0$))=1 | (FLAG($B_1B_0$)) = 1 |
| FBF | 0 0 1 0 | 0 0 $B_1B_0$ | 1 | 2½~4 | skip if (FLAG($B_1B_0$))=0 | (FLAG($B_1B_0$)) = 0 |
| CM | 0 0 0 0 | 1 1 0 0 | 1 | 1½~3 | skip if (Acc)=((DP)) | (Acc)=((DP)) |
| CI | 0 0 0 1 1 1 0 0 | 0 1 1 1 $I_3I_2I_1I_0$ | 2 | 2½~4 | skip if (Acc)=$I_3I_2I_1I_0$ | (Acc) = $I_3I_2I_1I_0$ |
| CLI | 0 0 0 1 1 1 1 0 | 0 1 1 0 $I_3I_2I_1I_0$ | 2 | 2½~4 | skip if $(DP_L)=I_3I_2I_1I_0$ | $(DP_L)$ = $I_3I_2I_1I_0$ |
| TC | 0 0 0 0 | 0 1 0 0 | 1 | 1½~3 | skip if (C)=1 | (C)=1 |
| TTM | 0 0 0 0 | 0 1 0 1 | 1 | 1½~3 | skip if (TMF/F)=1 | (TMF/F)=1 |
| TIT | 0 0 0 0 | 0 0 1 1 | 1 | 1½~3 | skip if (INTF/F)=1 INTF/F←0 | (INTF/F)=1 |
| JCP | 1 1 $P_5P_4$ | $P_3P_2P_1P_0$ | 1 | 1 | PC←$P_5$~$P_0$ | |
| JMP | 1 0 1 0 $P_7P_6P_5P_4$ | 0 $P_{10}P_9P_8$ $P_3P_2P_1P_0$ | 2 | 2 | PC←$P_{10}$~$P_0$ | |

Mnemonic. Instruction Code. Bytes. Machine Cycles. Operation. Condition for Skip.

| | $D_7D_6D_5D_4$ | $D_3D_2D_1D_0$ | | | | |
|---|---|---|---|---|---|---|
| JPA | 0 1 0 0 | 0 0 0 1 | 1 | 2 | PC←$A_3A_2A_1A_0$00 | |
| EI | 0 0 1 1 | 0 0 0 1 | 1 | 1 | INTE F/F←1 | |
| DI | 0 0 0 0 | 0 0 0 1 | 1 | 1 | INTE F/F←0 | |
| CZP | 1 0 1 1 | $P_3P_2P_1P_0$ | 1 | 1 | (PC)→STACK<br>PC←00000$P_3P_2P_1P_0$00 | |
| CAL | 1 0 1 0<br>$P_7P_6P_5P_4$ | 1$P_{10}P_9P_8$<br>$P_3P_2P_1P_0$ | 2 | 2 | (PC)→STACK<br>PC←$P_{10}$~$P_0$ | |
| RT | 0 1 0 0 | 1 0 0 0 | 1 | 2 | PC←(STACK) | |
| RTS | 0 1 0 0 | 1 0 0 1 | 1 | 3–4 | PC←(STACK)<br>PC←(PC)+1, 2 | Uncondi-tional |
| SEB | 0 1 1 1 | 0 1 $B_1B_0$ | 1 | 2 | PORT E($B_1B_0$)←1 | |
| REB | 0 1 1 0 | 0 1 $B_1B_0$ | 1 | 2 | PORT E($B_1B_0$)←0 | |
| SPB | 0 1 1 1 | 0 0 $B_1B_0$ | 1 | 1 | PORT ($DP_L$, $B_1B_0$)←1 | |
| RPB | 0 1 1 0 | 0 0 $B_1B_0$ | 1 | 1 | PORT ($DP_L$, $B_1B_0$)←0 | |
| TPA | 0 1 0 1 | 0 1 $B_1B_0$ | 1 | 2/3~4 | skip if (PORT A ($B_1B_0$))=1 | (PORT A ($B_1B_0$))=1 |
| TPB | 0 1 0 1 | 0 0 $B_1B_0$ | 1 | 1/2~3 | skip if (PORT ($DP_L$, $B_1B_0$))=1 | (PORT ($DP_L$, $B_1B_0$))=1 |
| OE | 0 1 0 0 | 0 1 0 0 | 1 | 2 | PORT E←(Acc) | |
| OP | 0 0 0 0 | 1 1 1 0 | 1 | 1 | PORT ($DP_L$)←(Acc) | |
| OCD | 0 0 0 1<br>$I_7I_6I_5I_4$ | 1 1 1 0<br>$I_3I_2I_1I_0$ | 2 | 2 | PORT C, D←$I_7$~$I_0$ | |
| IA | 0 1 0 0 | 0 0 0 0 | 1 | 2 | (PORT A)→Acc | |
| IP | 0 0 1 1 | 0 0 1 0 | 1 | 1 | (PORT ($DP_L$))→Acc | |
| STM | 0 0 0 1<br>1 0 $I_5I_4$ | 0 1 0 0<br>$I_3I_2I_1I_0$ | 2 | 2 | TMF/F←0<br>TIMER←$I_5$~$I_0$ | |
| NOP | 0 0 0 0 | 0 0 0 0 | 1 | 1 | No Operation | |

ELECTRICAL SPECIFICATIONS -- μPD546C

ABSOLUTE MAXIMUM RATINGS

Operating Temperature ..... -10°C to +70°C
Storage Temperature ..... -40°C to +125°C
Supply Voltage ..... -15 to +0.3 Volts
Input Voltages ..... -15 to +0.3 Volts
Output Voltages ..... -15 to +0.3 Volts
Output Current ..... -4mA

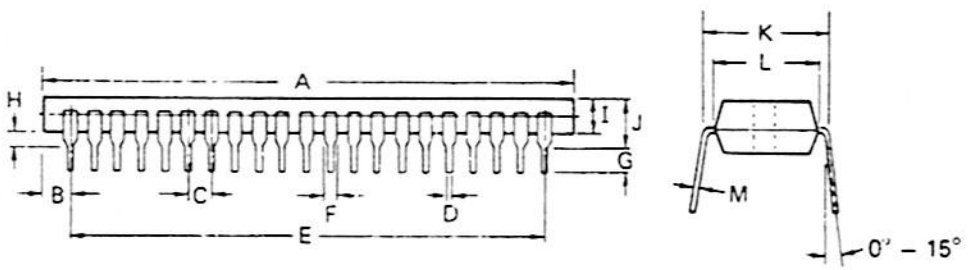
DC/AC CHARACTERISTICS
(Ta = -10°C to +70°C, $V_{GG}$ = -10V ± 10%)

| PARAMETER | SYMBOL | MIN. | TYP. | MAX. | UNIT | TEST CONDITIONS |
|---|---|---|---|---|---|---|
| Input High Voltage | $V_{IH}$ | 0 | | -2.0 | V | Ports A to D, $\overline{INT}$, RES |
| Input Low Voltage | $V_{IL}$ | -4.3 | | $V_{GG}$ | V | Ports A to D, $\overline{INT}$, RES |
| Input Leakage Current High | $I_{LIH}$ | | | +10 | μA | Ports A&B, $\overline{INT}$, RES, TEST $V_I$ = -1V |
| Input Leakage Current Low | $I_{LIL}$ | | | -10 | μA | Ports A&B, $\overline{INT}$, RES, TEST $V_I$ = -11V |
| I/O Leakage Current High | $I_{IOH}$ | | | +30 | μA | Ports C&D $V_I$ = -1V |
| I/O Leakage Current Low | $I_{IOL}$ | | | -30 | μA | Ports C&D $V_I$ = -11V |
| Output Voltage | $V_{OH}$ | | | -1.0 | V | Ports C to I $I_{OH}$ = -1.0mA |
| | $V_{OH}$ | | | -2.3 | V | Ports C to I $I_{OH}$ = -3.3mA |
| Output Leakage Current | $I_{OL}$ | | | -30 | μA | Ports C to I $V_O$ = -11V |
| Supply Current | $I_{GG}$ | | -30 | -50 | mA | |
| Oscillator Frequency | F | 150 | | 440 | KHz | |

CAPACITANCE (Ta=25°C, f=1 MHz)

| Parameter | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| Input Capacitance | Ci | | | 15 | pF |
| Output Capacitance | Co | | | 15 | pF |
| Input/Output Capacitance | Cio | | | 15 | pF |

PACKAGE OUTLINE




| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 56.0 MAX. | 2.2 MAX. |
| B | 2.6 MAX. | 0.1 MAX. |
| C | 2.54 | 0.1 |
| D | 0.5 ± 0.1 | 0.02 +0.004 −0.005 |
| E | 50.8 | 2.0 |
| F | 1.5 | 0.059 |
| G | 2.54MIN. | 0.1MIN. |
| H | 0.5 MIN. | 0.02 MIN. |
| I | 5.22 MAX. | 0.20 MAX. |
| J | 5.72 MAX. | 0.22 MAX. |
| K | 15.24 | 0.6 |
| L | 13.2 | 0.52 |
| M | 0.25 ± 0.1 | 0.01 ± 0.004 |

ELECTRICAL SPECIFICATIONS -- μPD553C

ABSOLUTE MAXIMUM RATINGS

Operating Temperature ..... -10°C to +70°C
Storage Temperature ..... -40°C to +125°C
Supply Voltage ..... -15 to +0.3 Volts
Input Voltages ..... -40 to +0.3 Volts
Output Voltages ..... -40 to +0.3 Volts
Output Current (Each Output Bit) ..... -12mA
(Total Current) ..... -55mA

DC/AC CHARACTERISTICS
(Ta = -10°C to +70°C, $V_{GG}$ = -10V ± 10%)

| PARAMETER | SYMBOL | MIN. | TYP. | MAX. | UNIT | TEST CONDITIONS |
|---|---|---|---|---|---|---|
| Input High Voltage | $V_{IH}$ | 0 | | -3.5 | V | Ports A to D, $\overline{INT}$, RES |
| Input Low Voltage | $V_{IL}$ | -7.5 | | | V | Ports A to D, $\overline{INT}$, RES |
| Input Leakage Current High | $I_{LIH}$ | | | +10 | μA | Ports A&B, $\overline{INT}$, RES, TEST $V_I$ = -1V |
| Input Leakage Current Low | $I_{LIL}$ | | | -10 | μA | Ports A&B, $\overline{INT}$, RES, TEST $V_I$ = -11V |
| | $I_{LIL}$ | | | -30 | μA | Ports A&B $V_I$ = -35V |
| I/O Leakage Current High | $I_{IOH}$ | | | +10 | μA | Ports C&D $V_I$ = -1V |
| I/O Leakage Current Low | $I_{IOL}$ | | | -10 | μA | Ports C&D $V_I$ = -11V |
| | $I_{IOL}$ | | | -30 | μA | Ports C&D $V_I$ = -35V |
| Output Voltage | $V_{OH}$ | -2.0 | | | V | Ports C to I $I_O$ = -8mA |
| Output Leakage Current | $I_{OL}$ | | | -10 | μA | Ports C to I $V_O$ = -11V |
| | $I_{OL}$ | | | -30 | μA | Ports C to I $V_O$ = -35V |
| Supply Current | $I_{GG}$ | | -30 | | mA | |
| Oscillator Frequency | F | | 400 | | KHz | |

CAPACITANCE
(Ta = 25°C, f = 1 MHz)

| PARAMETER | SYMBOL | MIN. | TYP. | MAX. | UNIT | TEST CONDITIONS |
|---|---|---|---|---|---|---|
| Input Capacitance | $C_I$ | | | 15 | pf | |
| Output Capacitance | $C_O$ | | | 15 | pf | |
| Input/Output Capacitance | $C_{IO}$ | | | 15 | pf | |

# EVACHIP-43
# (uPD556D)

## DESCRIPTION

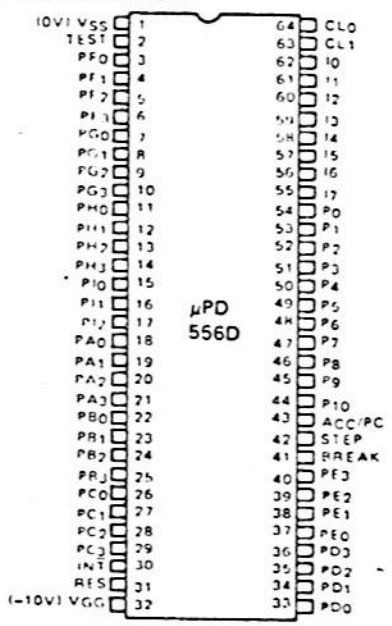
The uPD556D is an evaluation chip for the uCOM-43/44/45 single chip microcomputers. Designed to be used for both hardware and software debugging, the EVACHIP-43 is functionally equivalent to the uCOM-43, except that it does not contain on-chip ROM. Instead, it is able to address external memory. In addition, in order to facilitate debugging, the uPD556 is capable of displaying the contents of the internal accumulator and data pointer and of being single stepped.

When the uPD556 is being used to evaluate uCOM-44/45 designs, the external memory capacity should be restricted to that of the respective on-chip ROM and the instructions used should be restricted to the 58 comprising the uCOM-44/45 instruction set.

## FEATURES

- 4-bit parallel processor
- Full 80 instruction set of uCOM-43
- 10us instruction cycle
- Capable of addressing 2K x 8-bits of external program memory
- Single step capability
- Full functionality of uCOM-43
- Single supply, -10V PMOS technology
- 64 pin ceramic dual-in-line package

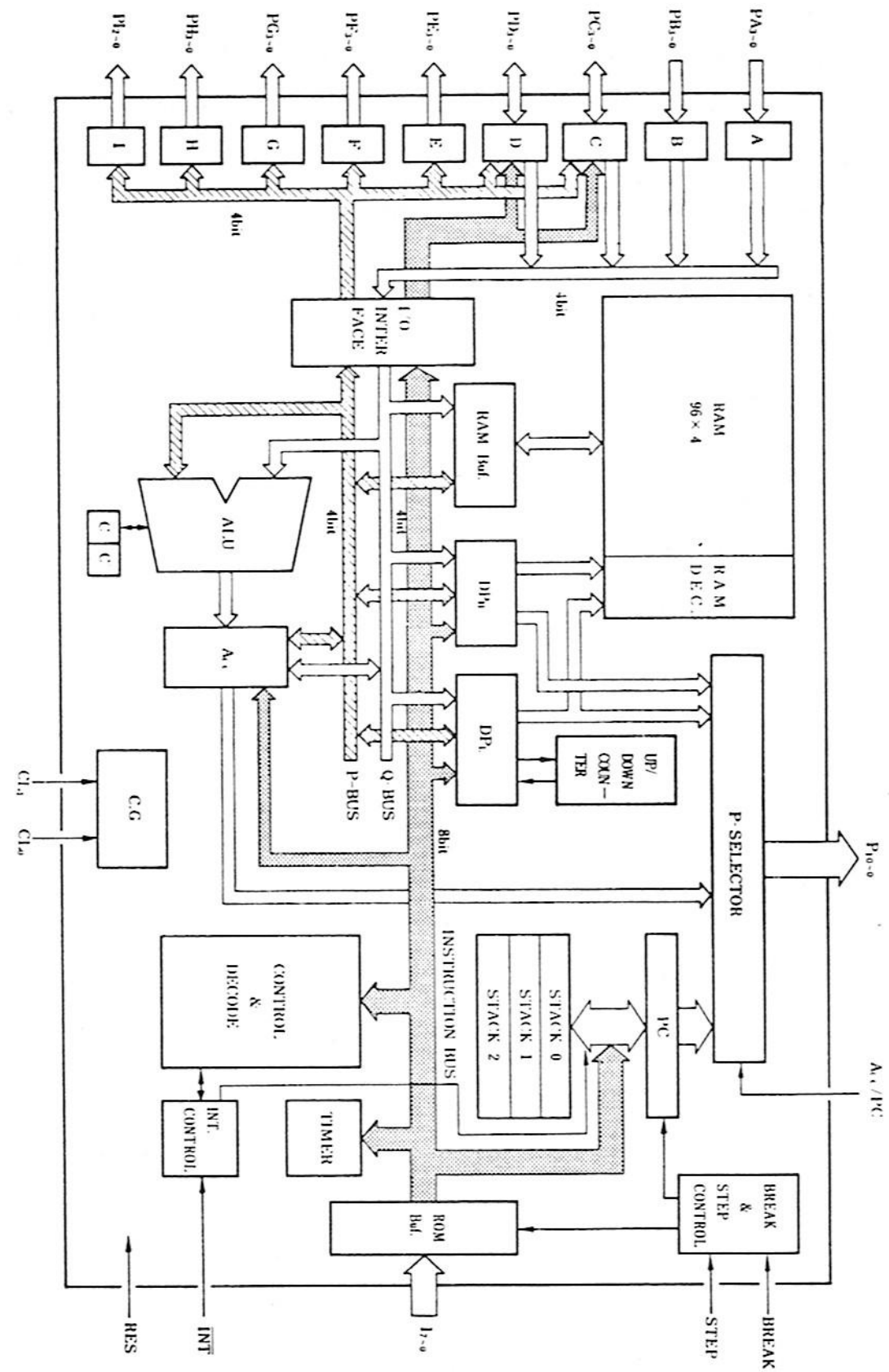
## PIN CONFIGURATION




PIN NAMES

| | |
|---|---|
| $PF_0 - PF_3$ | Output Port F |
| $PG_0 - PG_3$ | Output Port G |
| $PH_0 - PH_3$ | Output Port H |
| $PI_0 - PI_2$ | Output Port I |
| $PA_0 - PA_3$ | Input Port A |
| $PB_0 - PB_3$ | Input Port B |
| $PC_0 - PC_3$ | Input/Output Port C |
| $\overline{INT}$ | Interrupt Input |
| RES | Reset |
| $PD_0 - PD_3$ | Input/Output Port D |
| $PE_0 - PE_3$ | Output Port E |
| BREAK | Hold Input |
| STEP | Single Step Input |
| $A_{CC}/PC$ | Display $A_{CC}/PC$ Input |
| $P_0 - P_{10}$ | PC Output |
| $I_0 - I_7$ | Instruction Input |
| $CL_0 - CL_1$ | External Clock Source |

BLOCK DIAGRAM



ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS (Ta=25°C)

| Parameter | Symbol | Condition | Limits | Unit |
|---|---|---|---|---|
| Supply Voltage | $V_{GG}$ | | -15 to +0.3 | V |
| Input Voltage | $V_I$ | | -15 to +0.3 | V |
| Output Voltage | $V_O$ | | -15 to +0.3 | V |
| Output Current | $I_{OH}$ | All output pins. | -4 | mA |
| Operating Temperature | Topt | | -10 to + 70 | °C |
| Storage Temperature | Tstg | | -40 to +125 | °C |

D.C. CHARACTERISTICS(*)
(Ta=-10 to +70°C, $V_{GG}$=-10V ± 10%)

| Parameter | Symbol | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Input High Voltage | $V_{IH}$ | Port A to D $I_7$ to $I_0$, BREAK, STEP, $\overline{INT}$, RES, and ACC/PC | 0 | | -2.0 | V |
| Input Low Voltage | $V_{IL}$ | Port A to D $I_7$ to $I_0$ BREAK, STEP, $\overline{INT}$, RES, and ACC/PC | -4.3 | | $V_{GG}$ | V |
| Clock High Voltage | $V_{\phi H}$ | CLO Input | 0 | | -0.8 | V |
| Clock Low Voltage | $V_{\phi L}$ | CLO Input | -6.0 | | $V_{GG}$ | V |
| Input Leakage Current High | $I_{LIH}$ | Port A&B, $I_7$ to $I_0$ $\overline{INT}$, RES, BREAK, STEP ACC/PC, $V_I$=-1V | | | +10 | µA |
| | | Port C&D, $V_I$=-1V | | | +30 | µA |
| Input Leakage Current Low | $I_{LIL}$ | Port A&B, $I_7$ to $I_0$ $\overline{INT}$, RES, BREAK, STEP ACC/PC $V_I$=-11V | | | -10 | µA |
| | | Port C&D, $V_I$=-11V | | | -30 | µA |
| Clock Input Leakage High | $I_{L\phi H}$ | CLO Input, $V_{\phi H}$=0V | | | +200 | µA |
| Clock Input Leakage Low | $I_{L\phi L}$ | CLO Input, $V_{\phi L}$=-11V | | | -200 | µA |
| Output High Voltage | $V_{OH1}$ | Port C to I, $P_{10}$ to $P_0$ $I_{OH}$=-1.0mA | | | -1.0 | V |
| | $V_{OH2}$ | Port C to I, $P_{10}$ to $P_0$ $I_{OH}$=-3.3mA | | | -2.3 | V |
| Output Leakage Current Low | $I_{LOL}$ | Port C to I, $P_{10}$ to $P_0$ $V_O$=-11V | | | -30 | µA |
| Supply Current | $I_{GG}$ | | | -30 | -50 | mA |

(*) Relative to $V_{SS}$=0V.

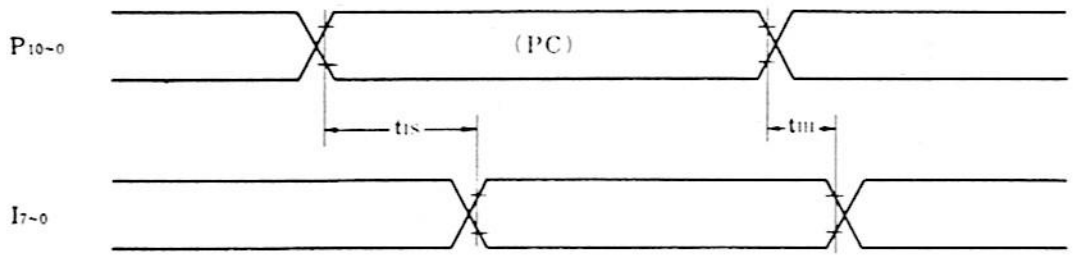
A.C. CHARACTERISTICS
(Ta=-10 to +70°C, $V_{GG}$=-10V ± 10%)

| Parameter | Symbol | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Frequency | f | | 150 | | 440 | KHz |
| Clock Rise & Fall Times | tr,tf | | 0 | | 0.3 | µs |
| Clock Pulse Width High | tØWH | | 0.5 | | 5.6 | µs |
| Clock Pulse Width Low | tØWL | | 0.5 | | 5.6 | µs |
| Input Setup Time | tIS | | | | 5 | µs |
| Input Hold Time | tIH | | 0 | | | µs |
| BREAK to STEP Interval | tBS | | 80 | | | tcy |
| STEP to RUN Interval | tSB | | 80 | | | tcy |
| STEP Pulse Width | tWS | | 12 | | | tcy |
| BREAK to ACC Interval | tBA | | 80 | | | tcy |
| ACC/PC Pulse Width | tWA | | 12 | | | tcy |
| STEP to ACC Interval | tSA1 | | 80 | | | tcy |
| PC to STEP Overlap | tSA2 | | | | 2 | tcy |
| PC to RUN Interval | tAB | | 0 | | | µs |
| ACC/PC→$P_{1C}$-$P_0$ Delay | tDAP1<br>tDAP2 | | | | 6<br>6 | tcy<br>tcy |

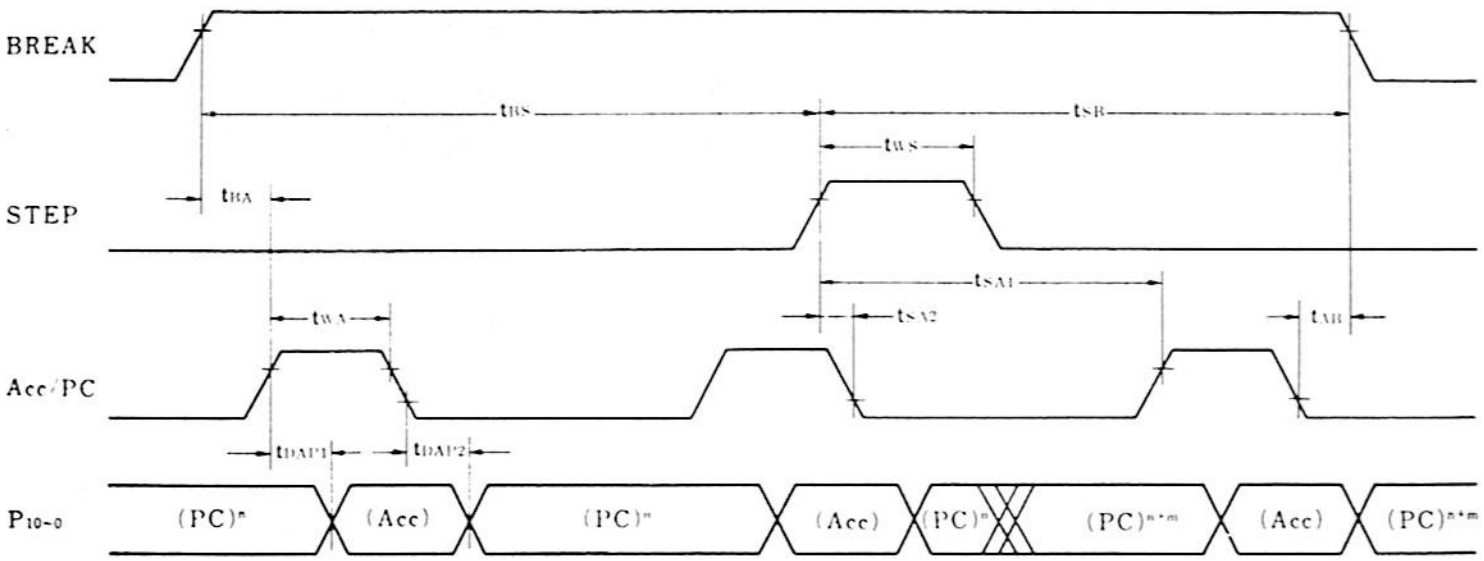
CAPACITANCE
(Ta=25°C)

| Parameter | Symbol | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Input Capacitance | $C_I$ | f = 1MHz | | | 15 | pF |
| Output Capacitance | $C_O$ | | | | 15 | pF |
| Input/Output Capacitance | $C_{IO}$ | | | | 15 | pF |

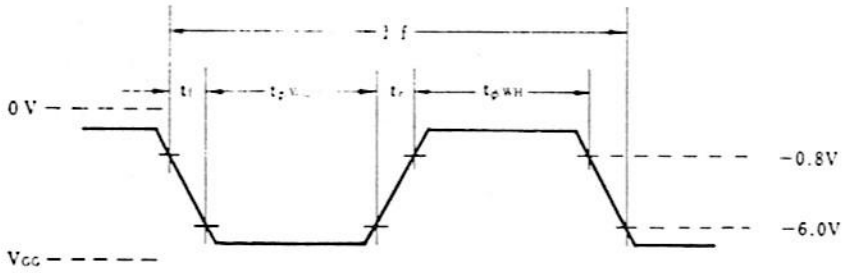
## TIMING WAVEFORMS



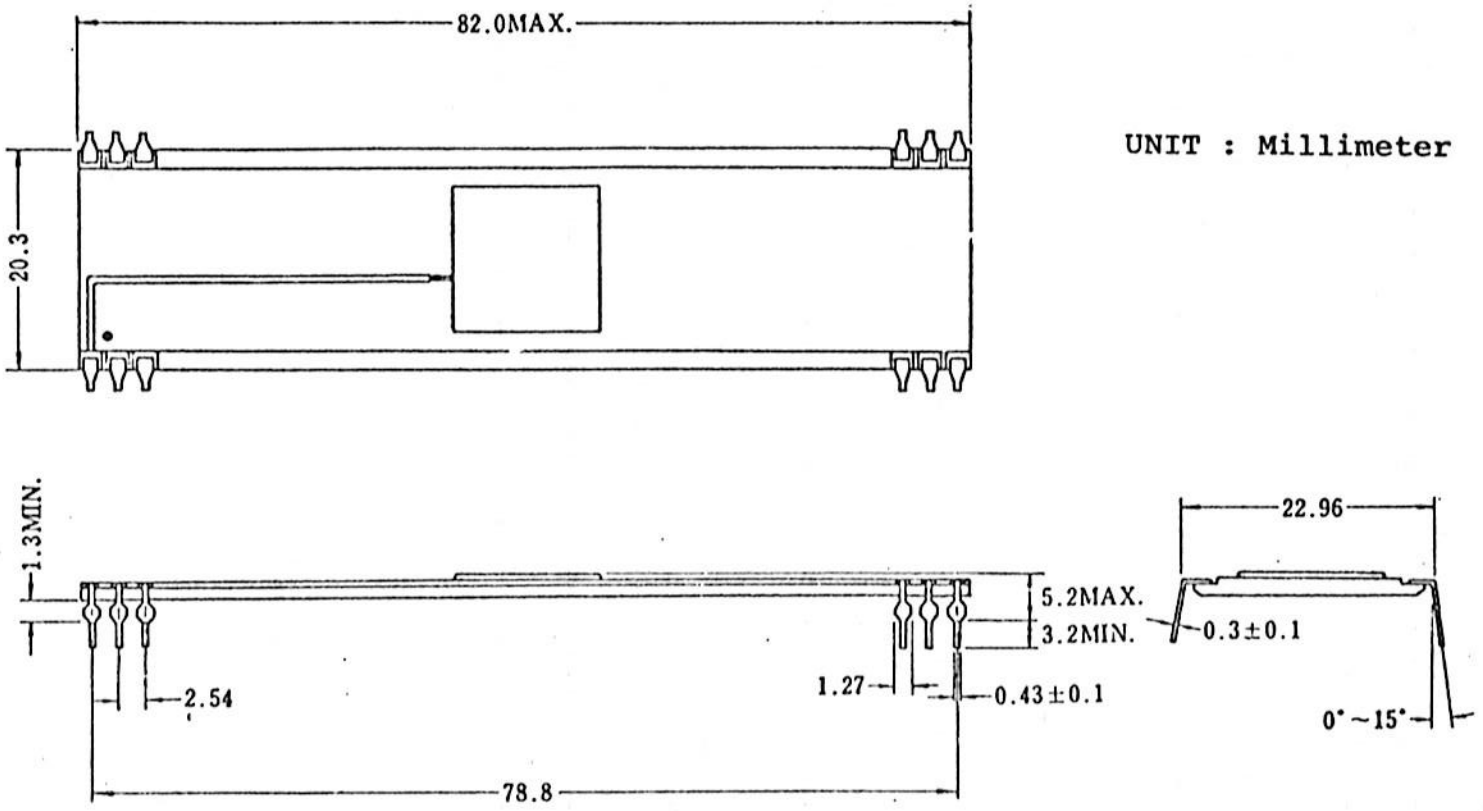

STEP, Acc/PC




## CLOCK WAVEFORMS



PACKAGE OUTLINE




UNIT : Millimeter

NEC microcomputers, inc.

Five Militia Drive
Lexington, Massachusetts 02173
Telephone (617) 862-6410
Telex 92-3434 TWX 710-326-6520

PRINTED IN U.S.A.

UM546-7-77-500
$15

NEC

NEC Microcomputers, Inc.

SUPPLEMENT TO

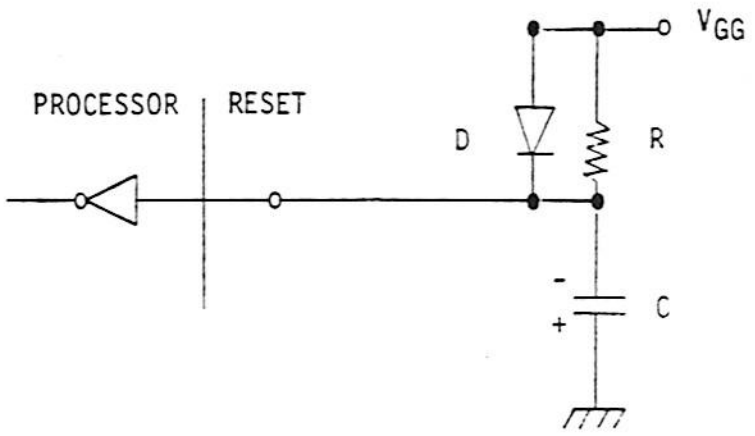
# μCOM-43
# SINGLE CHIP MICROCOMPUTER
## USERS' MANUAL

173 Worcester Street/Wellesley, Massachusetts 02181 Telephone 617/237-1910 TWX: 710-383-1745

# µCOM-43 ADDENDUM

## CORRECTIONS TO USERS' MANUAL

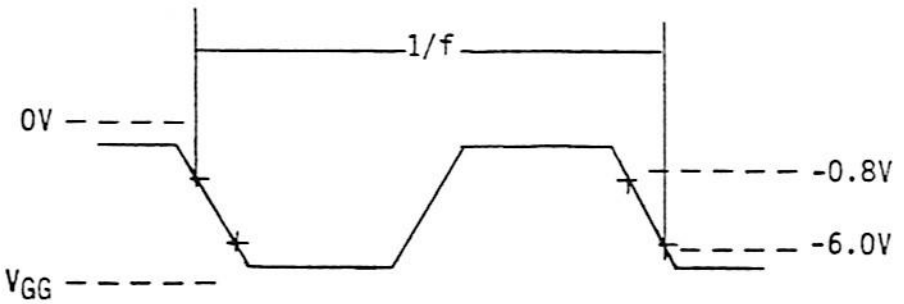
1) Power On Reset Circuit (Page 26)




2) CZP Instruction (Page 8)

The CZP instruction provides a subroutine call to one of 16 fixed addresses in field Ø, page Ø.

3) Clock Waveform



## ELECTRICAL SPECIFICATIONS -- μPD650C

### ABSOLUTE MAXIMUM RATINGS ($T_a$=25 C)

Operating Temperature ..... -30 C to 85 C
Storage Temperature ..... -55 C to 125 C
Supply Voltage ..... -0.3V to 7.0V
Input Voltages ..... -0.3V to 7.0V
Output Voltages ..... -0.3V to 7.0V
Output Current (Each Output Bit) ..... 2.5mA

### DC/AC CHARACTERISTICS
($T_a$ = -30 C to 85 C, $V_{CC}$ = 5V ± 10%)

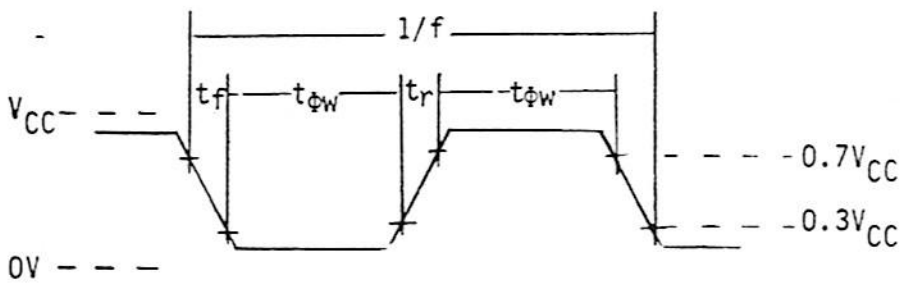
| PARAMETER | SYMBOL | MIN. | TYP. | MAX. | UNIT | TEST CONDITIONS |
|---|---|---|---|---|---|---|
| Input High Voltage | $V_{IH}$ | $0.7V_{CC}$ | | $V_{CC}$ | V | Ports A to D, $\overline{INT}$, RES |
| Input Low Voltage | $V_{IL}$ | 0 | | $0.3V_{CC}$ | V | Ports A to D, $\overline{INT}$, RES |
| Input Leakage Current High | $I_{LIH}$ | | | +10 | μA | Ports A&B, $\overline{INT}$, RES ($V_I$=$V_{CC}$) |
| Input Leakage Current Low | $I_{LIL}$ | | | -10 | μA | Ports A&B, $\overline{INT}$, RES ($V_I$=0V) |
| I/O Leakage Current High | $I_{IOH}$ | | | +10 | μA | Ports C&D ($V_I$=$V_{CC}$) |
| I/O Leakage Current Low | $I_{IOL}$ | | | -10 | μA | Ports C&D ($V_O$=$0_V$) |
| Output High Voltage 1 | $V_{OH1}$ | $V_{CC}$-0.5<br>$V_{CC}$-0.5 | | | V<br>V | Ports C&D ($I_{OH}$=-1mA)<br>Ports E&I ($I_{OH}$=-0.6mA |
| Output High Voltage 2 | $V_{OH2}$ | $V_{CC}$-2.5 | | | V | Ports C to I ($I_{OH}$=-2mA |
| Output Low Voltage | $V_{OL}$ | | | 0.6 | V | Ports E to I($I_{OL}$=2mA) |
| Supply Current | $I_{CC}$ | | 0.8 | 2.0 | mA | |
| Clock High Voltage | $V_{\Phi H}$ | $0.7V_{CC}$ | | $V_{CC}$ | V | CLO, Ext. Clk. |
| Clock Low Voltage | $V_{\Phi L}$ | 0 | | $0.3V_{CC}$ | V | CLO, Ext. Clk. |
| Clock Leakage Current High | $I_{L\Phi H}$ | | | 200 | μA | CLO, Ext. Clk. ($V_{OH}$=$V_{CC}$) |
| Clock Leakage Current Low | $I_{L\Phi L}$ | | | -200 | μA | CLO, Ext. Clk. ($V_{OL}$=0V) |
| Clock Frequency | f | 150 | | 440 | KHz | |

| PARAMETER | SYMBOL | MIN. | TYP. | MAX. | UNIT | TEST CONDITIONS |
|---|---|---|---|---|---|---|
| Clock Rise and Fall Times | tr,tf | 0 | | 0.3 | μS | Ext. Clk. |
| Clock Pulse Width | $t_{\Phi w}$ | 0.5 | | 5.6 | μS | Ext. Clk. |

CAPACITANCE
($T_a$ = -30 C to 85 C, $V_{CC}$ + 5V ± 10%)

| PARAMETER | SYMBOL | MIN. | TYP. | MAX. | UNIT | TEST CONDITIONS |
|---|---|---|---|---|---|---|
| Input Capacitance | $C_I$ | | | 15 | pf | |
| Output Capacitance | $C_O$ | | | 15 | pf | |
| I/O Capacitance | $C_{IO}$ | | | 15 | pf | |

CLOCK WAVEFORM



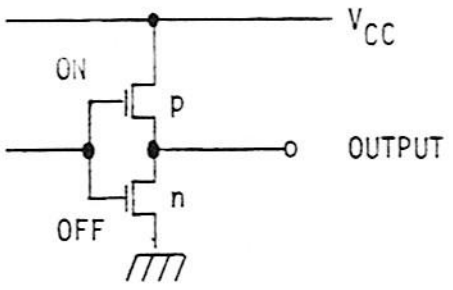
# OUTPUT BUFFER DESIGN

1) Output Ports (Ports E to I)

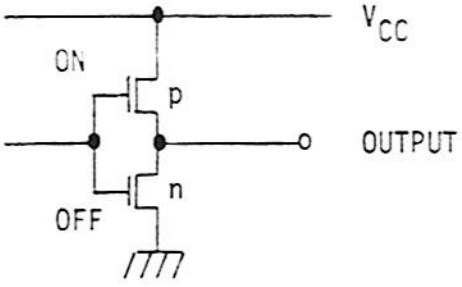
Push-Pull Output

Output Level Goes "Low" Upon Reset

High Level Output




Low Level Output



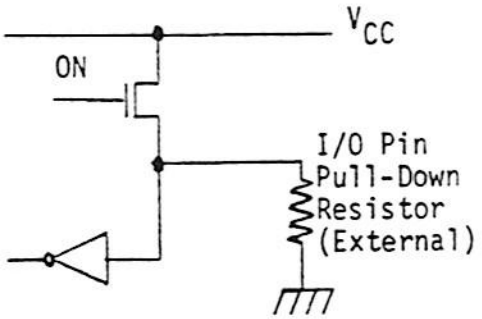

2) Input/Output Ports (Ports C and D)

P-Channel Open Drain Output

When Input State, Output TR is "OFF" ("0" Output).

Output TR is "OFF" Upon Reset

High Level Output




Input State