

# Using MCU Indirect Addressing

D/N : HA0006E

## Introduction

This document aims to introduce and describe how to use indirect addressing in clearing the internal memory RAM.

## Operating Description

Holtek MCUs have an indirect addressing register R0 ([00H]), (some MCUs have two indirect addressing registers R0 ([00H]) and R1 ([02H]), such as HT48R50A-1 and HT48R70A-1). MP0 (MP1) corresponds directly to the R0 (R1) pointer. In fact, any action on the R0 (R1) will result in corresponding read/write operations to the memory location specified by the corresponding MP0 (MP1). Therefore, read/write operation to R0 (R1) is equal to read/write operation to MP0 (MP1).

For example

```
MOV    A,20H
MOV    MP0,A           ;set MP0 direct to RAM address 20H
MOV    A,99H
MOV    [00H],A        ;move 99H to address 20H
```

This example is via R0, MP0, moving address 99H to address 20H

```
MOV    A,30H
MOV    MP1,A           ;set MP1 direct to RAM address 30H
MOV    A,88H
MOV    [02H],A        ;move 88H to address 30H
```

This example is via R1, MP1, moving 88H to MP0 direct to address 30H.

For Holtek MCU, a bank consist of 256 words and MP0 can only point to Bank0, in other words from 00H~FFH, then MP1 can point to any memory Bank (such as Holtek's HT47 and HT49 MCU series, which can use R1 (MP1) for BANK1 LCD indirect addressing register).

## Program Example

This program, through the manipulation of R0, MP0 shows how to clear the RAM contents. Clear the contents from 40H to 60H in Bank0, a total of 20H addresses.

```

r0     equ     [00h]
mp0    equ     [01h]
acc    equ     [05h]
code   .section at 0 'code'
        org 00h
        jmp start
        org 10h
start:
        mov     a,      40h
        mov     mp0,a    ;set the MP0 to point to the general
                        ;purpose memory RAM
        mov     a,      20h ;clear all to "0", move RAM(20H addresses)
                        ;to the accumulator
loop:
        clr     [00h]    ;clear the contents of 40H to "0" as
                        ;specified by MP0
        inc     mp0      ;MP0 incremented by 1 to point to next
                        ;address (41H)
        sdz     acc
        jmp     loop     ;loop if ACC is not equal to 0
        jmp     $
        end

```