Using the MCU Look-up Table Instructions

D/N：HA0007E

Introduction

Before discussing the look up table instruction usage, it is usesful to examine the Program Memory or Rom structure. The Holtek MCU Program Memory or ROM is divided up into pages each of which contains 256 words per page. The complete ROM space will be divided into several successive pages. The current page refers to some certain program instruction that is in the ROM page. The last page refers to the last page of the ROM. In addition, some MCUs like the HTG21X0 series have an especially large Program Memory capacity, in which case they are divided into areas of 8K words known as BANKs, each bank having 32 pages in capacity. The whole Program Memory is then made up from several of these 8K Banks.

Any area in the Program Memory can be designated as a look up table. To access this data Holtek MCUs provides two look up instructions, ″TABRDC [m]″ and ″TABRDL [m]″. The symbol [m] represents a register such as the accumulator, any other register or data memory.

As for those MCUs that do not provide a high pointer TBHP register, for example MCUs like the HT48 series, the ″TABRDC[m] ″ instruction is used to obtain data in the current page. For those MCUs that provide a TBHP register, for example the HTG21X0 series, the table instructions do not have current page limitations and can therefore look up all ROM data. For all devices, the ìTABRDL [m]î instruction can only look up data at the last page of the ROM.

For devices with no high table pointer TBHP register, it is first necessary to setup the TBLP register before the look-up instructions are executed. For devices which have a TBHP register both this register and the TBLP register must be setup before the ″TABRDC [m] ″ instruction is executed. When using ″TABRDL [m] ″, TBHP will be taken as the highest address of the last page.

When the look up table instruction is executed, the low byte of the table data will be transferred to [m] while the high will be tranferred to the TBLH regiser. As the TBLH regiser is a read only register, the problem of executing a look up table instruction in the main program and the interrupt service subroutine at the same time should be handled properly. To avoid such a situation, besides not using look up table simultaneously in the main and interrupt subroutine, ″RET A, X″ instruction can also be used as a look up table function. The following note will introduce the three kinds of look up table instructions as mentioned above.

Usage

″TABRDC [m]″ Instruction Usage

- For MCUs with no TBHP register for looking up the current page data, even if it is in the last page. For example:

```
#INCLUDE HT48R10A-1.INC
CODE .SECTION AT 000H 'CODE'
...
MOV     A,60H
MOV     TBLP,A              ; TBLP=60H (POINTOR)
TABRDC  ACC                 ; ACC=34H
                            ; NOP to examine THE LOW BYTE
MOV A,  TBLH                ; ACC=12H
NOP                         ; NOP to examine THE HIGH BYTE
...
ORG     60H
DC      1234H, 3567H        ; TABLE
...
```

- In the TBHP register MCUs for looking up all the ROM data. For example:

```
#INCLUDE HTG2190.INC
ROMBANK 0 CODE0
ROMBANK 1 CODE1
CODE0 .SECTION AT 0H 'CODE'
...
MOV A,3CH
MOV TBHP,A                  ; TBHP=3CH (HIGH POINTER)
MOV A,01H
MOV TBLP,A                  ; TBLP=01H (LOW POINTER)
TABRDC ACC                  ; ACC=45H
NOP                         ; NOP TO EXAMINE THE LOW BYTE
MOV A,TBLH                  ; ACC=23H
NOP                         ; NOP to examine THE HIGH BYTE
...
CODE1 .SECTION AT 0H 'CODE'
...
```

```
ORG     1C00H
DW      1234H, 2345H       ; TABLE
...
```

"TABRDL [m] " Instruction Usage (To Look-up Data at the Last Page of the ROM)

For example:

```
MOV     A,00H
MOV     TBLP,A             ; TBLP=00H (POINTER OR LOW POINTER)
TABRDL  ACC                ; ACC=77H
NOP                        ; NOP to examine THE LOW BYTE
MOV A,  TBLH               ; ACC=12H
NOP                        ; NOP to examine THE HIGH BYTE
...
ORG     XXXXH              ; THE FIRST ADDRESS OF LAST PAGE
DC      1277H, 3567H       ; TABLE
...
```

"RET A, X" Instruction Usage (Program Returning from Subroutine Will Send Immediate Value to Accumulator)

For example:

```
...
MOV  A,1         ; ACC IS A POINTER TO THE DATA IN TABLE
CALL TABLE       ; SIMULATE READING TABLE
NOP              ; ACC=77H
...
TABLE:
        ADDM    A,PCL      ; JUMP TO THE DATA
        RET     A,61H      ; RET,ACC=61H
        RET     A,77H      ; RET,ACC=77H
        RET     A,98H      ; RET,ACC=98H
        RET     A,18H      ; RET,ACC=18H
```

Note: Changing the value of the PCL does not permit jumping to different pages, therefore the TABLE subprogram must be also at the same page. In other words, when executing an ADDM A, PCL instruction, result overflow is not allowed, otherwise a program eorror will occur. Besides, only one byte can be returned in this method.

Appendices

Appndix 1：For Devices that do not Contain a TBHP Register, here the HT48R10A-1 is Taken as an Example

```
#INCLUDE HT48R10A-1.INC
CODE .SECTION AT 0 'CODE'
       ORG 000H
       JMP START
       ORG 004H
RETI
START:
       MOV     A,      60H
       MOV     TBLP,A              ; TBLP=60H (POINTER)
       TABRDC  ACC                 ; ACC=34H, TBLH=12H
       JMP $

       ORG 60H
       DC  1234H,3567H             ; TABLE(THE LENGTH OF WORD IN
                                   ; HT48R10A-1 IS 14.)
END
```

The table below shows the range of corresponding text length and data size of MCUs

MCU Part No.	Text Length	Data Size
HT48R10A-1	14	0000~3FFF
HT48R30A-1	14	0000~3FFF
HT48R50A-1	15	0000~7FFF
HT48R70A-1	16	0000~FFFF

Appendix 2：For MCUs that Contain a TBHP Register MCUs, here the HTG2190 is Taken as an Example

```
#INCLUDE HTG2190.INC
ROMBANK 0 CODE0
ROMBANK 1 CODE1
CODE0 .SECTION AT 0H 'CODE'
       ORG 0000H
       JMP START
       ORG 0020H
START：
       MOV A,3CH
       MOV TBHP,A                  ; TBHP=3CH (HIGH POINTER)
       MOV A,01H
       MOV TBLP,A                  ; TBLP=01H  (LOW POINTER)
       TABRDC ACC                  ; ACC=45H
       NOP                         ; NOP TO EXAMINE THE LOW BYTE
       MOV A,TBLH                  ; ACC=23H
```

```
        NOP                         ; NOP TO EXAMINE THE HIGH BYTE
        JMP $
CODE1 .SECTION AT 0H 'CODE'
        ORG 1C00H
        DW  1234H, 2345H            ; TABLE
END
```

Appendix 3：For all MCUs, here the HT48R10A-1 is Taken as an Example

```
#INCLUDE HT48R10A-1.INC
CODE .SECTION AT 0 'CODE'
        ORG 000H
        JMP START
        ORG 004H
        RETI
START：
        MOV     A,00H
        MOV     TBLP,A              ; TBLP=00H (POINTER OR LOW POINTER)
        TABRDL  ACC                 ; ACC=77H
        NOP                         ; NOP TO EXAMINE THE LOW BYTE
        MOV     A,TBLH              ; ACC=12H
        NOP                         ; NOP TO EXAMINE THE HIGH BYTE
        JMP $

        ORG 300H                    ; THE FIRST ADDRESS OF LAST PAGE
        DC  1277H, 3567H            ; TABLE
END
```

Note: HT48R10A-1 is used in this case. Due to different ROM sizes of each MCU, the address of the last page differs. To define the table at the last page, it is necessary to know the starting address of the last page.

The table below shows the range of the last page of each MCU (HT48 series)

MCU Part No.	Last Page Range
HT48R10A-1	300H~3FFH
HT48R30A-1	700H~7FFH
HT48R50A-1	F00H~FFFH
HT48R70A-1	1F00H~1FFFH

Appendix 4：also for all MCUs (Here we take HT48R10A-1)

```
CODE .SECTION AT 0 'CODE'
        ORG     0
        JMP     START
        ORG     4
        RETI
```

```
START:
      MOV  A,1                  ; ACC IS A POINTER TO THE DATA IN
TABLE
      CALL    TABLE             ; SIMULATE READING TABLE
      NOP                       ; ACC=77H
JMP   $

TABLE:
       ADDM   A,PCL             ; JUMP TO THE DATA
       RET    A,61H             ; RET,ACC=61H
       RET    A,77H             ; RET,ACC=77H
       RET    A,98H             ; RET,ACC=98H
       RET    A,18H             ; RET,ACC=18H
END
```