MOTOROLA

TECHNICAL UPDATE

MC68HC05P8

Technical Update contains updates to documented information appearing in other Motorola technical documents as well as new information not covered elsewhere.

We are confident that your Motorola product will satisfy your design needs. This Technical Update and the accompanying manuals and reference documentation are designed to be helpful, informative, and easy to use.

Should your application generate a question or a problem not covered in the current documentation, please call your local Motorola distributor or sales office. Technical experts at these locations are eager to help you make the best use of your Motorola product. As appropriate, these experts will coordinate with their counterparts in the factory to answer your questions or solve your problems. To obtain the latest document, call your local Motorola sales office.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

TABLE OF CONTENTS

MC68HC05P8 Modules

MC68HC05P8 Part Specific

TECHNICAL UPDATE

MC68HC05P8 Modules

Analog-to-Digital Converter Module

ATD4x8_A

Revision History

Date	Revision	Description
4/28/95	1.00	Original release. Includes tracker HC05P8.002.

Analog-to-Digital (A/D) Example Code

Reference Document: MC68HC05P8/D, page 8-1

Tracker Number: HC05P8.002 Revision: 1.00

The following code shows a simple routine that will read AN0 of the analog-to-digital (A/)D converter. Port D bit 0 is the A/D converter input AN0. The code was tested on an M68HC05P8EVS evaluation system. It is assumed the reader is familiar with the IASM05 assembler and the EVM05 debugger from P&E Microcomputer Systems. These P&E software programs are used to assemble, download, and run the code.

Assemble these lines of code:

```
****    Equates
ADSCR   EQU     $1E                 ;a/d status and ctrl reg
ADDR    EQU     $1D                 ;a/d data reg


        ORG     $1680               ;start of program

START   LDA     #$20
        STA     ADSCR               ;turn on the a/d converter

        LDA     #33T                ;execute loop for 100 usecs
WAIT    DECA                        ; so a/d can warm up
        BNE     WAIT
        LDA     #$20
        STA     ADSCR                  ;start a conversion
CHCK    LDA     ADSCR                  ;check if conversion
        AND     #$80                   ; complete flag (COCO)
        BNE     CHCK                   ; is set

        LDA     ADDR                   ;when conversion done
                                       ; put answer in ACCA

LOOP    BRA     LOOP                   ;loop forever

        ORG     $1FFE                  ;define reset vector
        DW      $1680
```

The connections needed for this routine are:

```
VRH  -------------------->   ---  +5 volts
                              |
                              /
                              \
Port D0, AN0  --------------> /   10K Pot
                              \
                              /
                              |
VRL --------------------->   ---  GND
```

The V_{RH} is connected to +5 volts. The V_{RL} is connected to ground. This gives the A/D conversion range a delta of 5 volts. The potentiometer (pot) will vary the voltage between V_{RH} and V_{RL}. This voltage also is fed into converter channel AN0.

To predict what the A/D converter will read, use this relationship:

$$\frac{\text{A/D reading}}{255} = \frac{\text{pot voltage on AN0}}{5\text{ volts}}$$

Steps to run the code:

1. Download the code into the EVS with the EVM05 software.
2. Set the program counter to $1680, PC 1680.
3. Type GO.
4. The routine will run and eventually hit the infinite loop. Press the ABORT button on the EVS to get out of the loop.
5. The A/D reading is in accumulator A. Accumulator A is shown on the EVM05 screen.

Some voltage inputs with their appropriate A/D readings are:

0.0 V --> $00

1.0 V --> $33

2.5 V --> $80

4.0 V --> $CC

5.0 V --> $FF

The reading may be off a couple of least significant bits (LSB) due to noise in the system.

Computer Operating Properly (COP)

COP0COPRT2_A

Revision History

Date	Revision	Description
4/28/95	2.00	Original release. Includes tracker HC705K1.005.

COP Timeout Test

Reference Document: Not applicable

Tracker Number: HC705K1.005 Revision: 2.00

The program below tests the timeout on the COP module COP0COPRT2. The program may be used on any HC05 as long as the part has the COP0COPRT2 module. Memory and reset vectors may need to be changed to work properly with a particular MCU. The HC705K1 was used to verify operation.

```
*****************************************************************
*
* Program Name: 7K1_COP.ASM ( COP Test on the HC705K1 )
* Revision: 1.00
* Date: May 25, 1993
*
* Written By: Mark Glenewinkel
*             Motorola CSIC Applications
*
* Assembled Under: P&E Microcomputer Systems IASM05
*
```

```
*       *********************************
*       *       Revision History        *
*       *********************************
*
*       Rev     1.00    03/26/93        M.R. Glenewinkel
*                       Initial Release
*
*****************************************************************
*
* Program Description:
*
*       This program is a simple routine that tests the COP
*       timeout on the HC705K1 MCU. The HC705K1 was programmed
*       with the M68HC705KICS board. The part was then tested
*       for COP resets on a protoboard. If the COP is working
*       correctly, PortA will toggle on approximately
*       1/2 sec intervals.
*
*****************************************************************

PORTA   equ     $00
DDRA    equ     $04
MOR     equ     $17

        ORG     MOR
        DB      $01             ;enable COP

        ORG     $200

START   lda     #$FF            ;make port A all output
        sta     DDRA            ;

        com     $E0             ;complement RAM mem $E0
        lda     $E0             ;ACCA <- ($E0)
        sta     PORTA           ;port A <- (ACCA)


DONE    NOP                     ;branch into an infinite loop
        BRA     DONE            ;waiting for a COP timeout

        ORG     $03FE           ;define reset vector
        DW      START
```

MC68HC05P8 Part Specific

Revision History

Date	Revision	Description
4/28/95	1.00	Original release. Includes trackers HC05P8.001, HC05P8.003, HC05P8.004, HC05P8.005, and HC05P8.006.

Analog-to-Digital (A/D) Accuracy Spec

Reference Document: MC68HC05P8 Databook, page 10-7

Tracker Number: HC05P8.001 Revision: 1.00

The accuracy spec of the A/D converter has caused some confusion. The spec reads:

Characteristic	Min	Max	Unit
Absolute Accuracy	-	+/- 1-1/2	LSB

Some customers have misread this as "plus or minus 1.0 LSB down to 0.5 LSBs."

However, the correct reading is "plus or minus 1.5 LSBs."

Maximum Stop Mode I_{DD} at V_{DD} = 5 V

Reference Document: MC68HC05P8/D, page 10-3

Tracker Number: HC05P8.003 Revision: 1.00

The following preliminary information is not a guaranteed spec:

The maximum stop mode I_{DD} at V_{DD} = 5.0 V is 300 µA. This value applies to all temperature ranges.

Maximum Stop Mode I_{DD} at V_{DD} = 3.3 V

Reference Document: MC68HC05P8/D, page 10-4

Tracker Number: HC05P8.004 Revision: 1.00

The following preliminary information is not a guaranteed spec:

The maximum stop mode I_{DD} at V_{DD} = 3.3 V is 250 µA. This value applies to all temperature ranges.

Stop Mode Application Example

Reference Document: MC68HC05P8/D, page 4-24

Tracker Number: HC05P8.005 Revision: 1.00

```
*******************************************************************************
*******************************************************************************
*                                                                             *
*                      STOP program example for HC05P8                    *
*                                                                             *
*******************************************************************************
*                                                                             *
* Program Name: STOPP8.ASM                                                    *
* Revision: 0.01                                                              *
* Date: 12/16/91                                                              *
* Written By: Robert Chretien & David Yoder                                   *
*             Motorola CMCU Applications                                      *
* Assembled Under: P&E Microcomputer Systems, Inc.  IASM05                    *
*                                                                             *
* Program Description:                                                        *
*             This progam shows how to use the MC68HC705P9 STOP               *
*             instruction. It is meant to be used in a stand a lone mode,     *
*             or with an appropriate evaluation/emulation system.             *
*                                                                             *
*             Upon executing the program, PA0 will toggle. When PA1           *
*             is pulled high, the MCU will enter STOP mode and PA0            *
*             will cease to toggle.                                           *
*             An external reset or an even on IRQ will cause the MCU          *
*             to exit from stop mode.                                         *
*                                                                             *
*                                                                             *
*               _________                                                     *
*               | START |                                                     *
*               ---------                                                     *
*                   |                                                         *
*          /--------|                                                         *
*          |        |                                                         *
*          |    __________                                                    *
*          |    | Toggle |                                                    *
*          |    |  PA0   |                                                    *
*          ^    ----------                                                    *
*          |        |                                                         *
*          |       / \                                                        *
*          |     /     \                                                      *
*          | n  /  PA1  \                                                     *
*          |---<  High ? >                                                    *
*          |    \       /                                                     *
*          |     \     /                                                      *
*          |       \ /                                                        *
*          ^        | y                                                       *
*          |    ---------                                                     *
*          |    | STOP  |                                                     *
*          |    ---------                                                     *
*          |        |                                                         *
*          \--------/                                                         *
*                                                                             *
*******************************************************************************
*                                                                             *
* Revision Descriptions:                                                      *
* 1. Rev 0.00: Original program.                                              *
* 2.                                                                          *
*                                                                             *
*                                                                             *
*******************************************************************************
*******************************************************************************
*
*
*
***************************************
* MCU Equates                         *
***************************************
PortA   equ   $0000
PortB   equ   $0001
DDRA    equ   $0004
DDRB    equ   $0005

***************************************
* Interrupt vectors                   *
***************************************
        org   $1FF8
TIMER   fdb   TRAP
IRQ     fdb   IRQISR
SWI     fdb   TRAP
RESET   fdb   START
```

```
***************************************
* Start                               *
* Main Loop of code                   *
***************************************
        ORG     $1680           ; Begin code in EPROM
Start   LDA     #$01
        STA     DDRA            ; Set port A0 to output, leave
                                ;  others as inputs

Toggle  LDA     PORTA           ; Toggle port A0. This will toggle
        EOR     #%00000001      ;  while the code is running.
        STA     PORTA           ;  This will stop toggling when STOP
                                ;  mode is entered. When STOP mode
                                ;  is exited with IRQ or RESET, this
                                ;  will resume toggling.

        LDA     PORTA           ; See if PA1 has been pulled high
        AND     #$02            ; If not, branch to TOGGLE to toggle
        BEQ     TOGGLE          ;  PA0 again.
                                ; If so, enter STOP mode.

        STOP                    ; Enter STOP mode.
                                ;  This will:
                                ;    Clear interrupt flag in status
                                ;     register - no need to do CLI for
                                ;     IRQ to exit from STOP
                                ;    Disable the oscilltor - you will
                                ;     see OSC2 stop toggling when STOP
                                ;     mode is successfully entered.

        BRA     TOGGLE          ; Stay in main loop toggling

***************************************
* IRQISR                              *
***************************************
* Service routine for                 *
* external interrupts                 *
*                                     *
* Does nothing, only returns to       *
* main routine.                       *
***************************************
IRQISR:
        NOP
        RTI

***************************************
* TRAP                                *
***************************************
* Routine for unused interrupts       *
*                                     *
* Traps in a branch to self           *
***************************************
TRAP    BRA     TRAP            ; Trap interrupts

        END
```

BSET and BCLR are RMW Instructions

Reference Document: MC68HC05P8/D, page number 4-15

Tracker Number: HC05P8.006 Revision: 1.00

The tables, showing instruction types, list BSET and BCLR as bit manipulation instructions only. While correct, this is not complete. These operations use a read-modify-write method to accomplish their task. Therefore, they should be included in table 4-8.

NOTE: These instructions do not use the same addressing modes as the other read-modify-write istructions. Only direct addressing is valid for BSET and BCLR.

Because BSET and BCLR are read-modify-write instructions, they may not be used with write-only registers. These registers will read back undefined data. Therefore, a read-modify-write operation will read undefined data, modify it as appropriate, and then write it back to the register. Because the original data is undefined, the data written back will be undefined.