AN1738

Instruction Cycle Timing of MC68HC05JJ/JP Series Microcontrollers

By Mark L. Shaw
Member, Technical Staff, Sensor Products Division
Transportation Systems Group
Phoenix, Arizona

Introduction

The MC68HC05JJ and MC68HC05JP (JJ/JP) series of microcontrollers has an asynchronous analog interface coupled to the digital CPU. Because of this, events can occur which are not specifically synchronized to the software that is running.

For example, when sampling the outputs of the two voltage comparators, the actual time when the CMP1 or CMP2 bits are read is dependent on bus speed and the instruction being executed. The user can determine the time and bus cycles of an instruction based on the oscillator frequency being used. The timing within an instruction is usually not known, although some assemblers provide the cycles per instruction or the time between instructions.

Normally, such timing for microcontrollers is not published, since this is not a consideration when all their peripherals are digital or have analog-to-digital (A/D) convertors which have defined sampling schemes. Also, the user must understand this instruction timing to be able to write software which properly measures the exact timing of the external ramping capacitor when doing A/D conversions in mode 0 or mode 1.

Information in this application note describes the hardware timing of the JJ/JP series and provides a method whereby the user can make individual timing measurements.

This measurement technique also can be applied to other members of the MC68HC05 Family of MCUs.

Typical Read Instruction

The typical timing diagram for a 3-cycle LDA instruction (direct addressing mode) is shown in **Figure 1**.

The important time for all instructions which read data is the rising edge of the internal bus clock during the read cycle. Even though the location may be enabled before the end of the cycle, it is the next rising edge of the internal bus clock (usually called PH2) where the data is latched into the CPU.

Not all instructions which read data will do so at the end of the instruction cycle in the sequence. For instance, instructions such as BRCLR, BRSET, or any of the read-modify-write instructions will read the data on an earlier cycle so that the data can be used in the later cycles of the instruction.

Also, there may be “dummy” read cycles where the CPU has pointed the address toward the target register, but has not actually accessed its data.

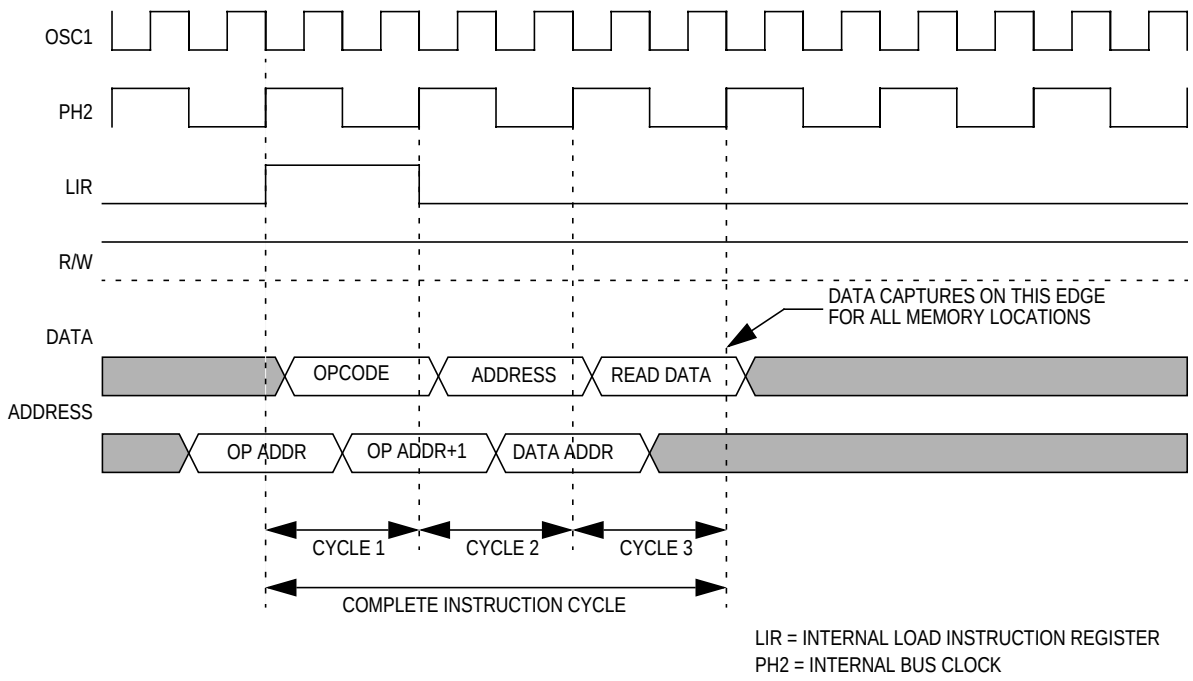

Figure 1. Typical Instruction Timing (LDA, Direct)

Typical Write Instruction

The typical timing diagram for a 4-cycle STA instruction (direct addressing mode) is shown in **Figure 2**.

The important time for all instructions which write data is the last cycle of the instruction sequence where the data is written into the target location. For most locations, like port B and the internal registers, the data is written at the middle of this last cycle. However, any writes to port A or port C will start at the beginning of the last instruction cycle. Essentially, the port B or internal registers are written about one-half cycle later in the instruction than port A or port C. The data may be presented to the target location for the complete write cycle, but the actual change in the data location will occur as soon as the data is presented to the location.

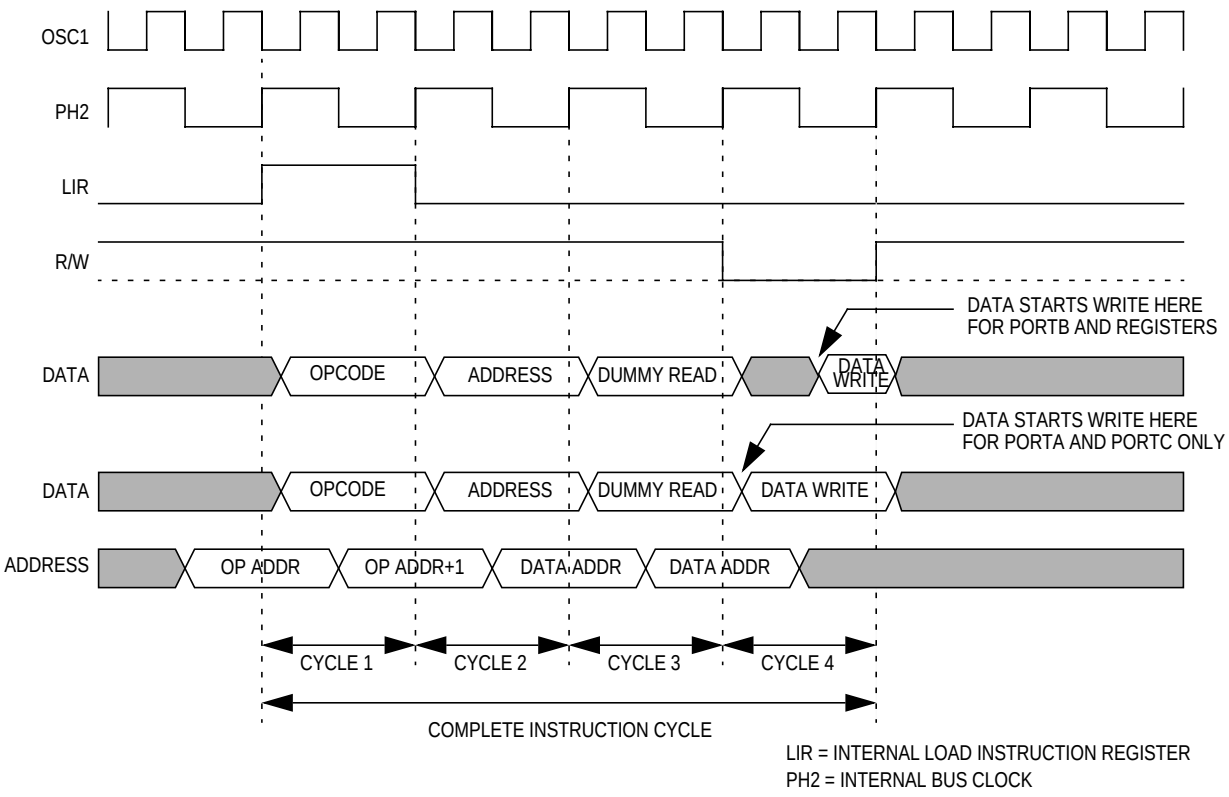

Figure 2. Typical Instruction Timing (STA, Direct)

The actual times at which the data is latched on a read cycle and the data is written on a write cycle may be up to 100 ns before or after the rising and/or falling edges of the internal PH2 bus clock.

This level of timing accuracy is not needed for several reasons:

- The user cannot see the PH2 clock externally.
- The highest bus frequency is 2.1 MHz (period = 476 ns).
- The core timer counts every four bus cycles (period = 1.9 seconds).
- The 16-bit timer counts every four bus cycles (period = 1.9 seconds).
- The fastest software polling loop is five bus cycles (2.4 seconds).

With these timing resolutions, a 100-ns variation is less than approximately 5% of a count. Better accuracy in measuring time is best done by relating the timing of events to the nearest one-half cycle from the first instruction.

Instruction Timing Summary

In Table 1 a summary of the read and write timing is given for the instructions most often used to manipulate or poll the registers and ports.

Table 2 gives similar timing information for the other instructions which can manipulate or read the registers or ports. These other instructions are only given for the direct addressing mode. Using the extended or indexed addressing modes will add one or two additional addressing cycles which will appear before the read or write cycles.

Table 1. Read/Write Timing for Common Instructions

Opcode	Instruction	Addressing mode	Total CPU cycles	Dummy read cycle	Timing edge from start of instruction (cycles)		
					Read (note 2)	Write (note 3)	Write (note 4)
BCLR0:BCLR7 BSET0:BSET7	Bit clear Bit set	Bit manipulation (note 1)	5	3	4	4	4.5
BRCLR:BRCLR7 BRSET:BRST7	Branch if bit clear Branch if bit set	Bit test and branch	5	—	3	—	—
LDA or LDX	Load accumulator or Load index register	Direct	3	—	3	—	—
		Extended	4	—	4	—	—
		Index, no offset	3	—	3	—	—
		Index, 8-bit offset	4	—	4	—	—
		Index, 16-bit offset	5	—	5	—	—
STA or STX	Store accumulator or Store index register	Direct	4	3	—	3	3.5
		Extended	5	4	—	4	4.5
		Index, no offset	4	3	—	3	3.5
		Index, 8-bit offset	5	4	—	4	4.5
		Index, 16-bit offset	6	5	—	5	5.5

1) BCLR and BSET are read-modify-write instructions.
2) Data read is accessed up until to the end of the read cycle.
3) Data is written at the start of the write cycle for port B and internal registers.
4) Data is written halfway through write cycle for port A and port C only.

Table 2. Read/Write Timing for Other Instructions

Opcode	Instruction	Addressing mode (note 1)	Total CPU cycles	Dummy read cycle	Timing edge from start of instruction (cycles)		
					Read (note 2)	Write (note 3)	Write (note 4)
ADC ADD AND	Add with carry Add w/o carry Logical AND	Direct	3	—	3	—	—
ASL ASR	Arith. shift left Arith. shift right	Direct*	5	3	4	4	4.5
BIH BIL	Branch, IRQ high Branch, IRQ low	Relative	3	—	2	—	—
BIT	Bit test	Direct	3	—	2	—	—
CLR	Clear	Direct*	5	3	4	4	4.5
CMP CPX	Compare accumulator Compare X-reg	Direct	3	3	3	—	—
COM	Complement	Direct*	5	3	4	4	4.5
DEC	Decrement	Direct*	5	3	4	4	4.5
EOR	Exclusive OR	Direct	3	—	3	—	—
INC	Increment	Direct*	5	3	4	4	4.5
LSL LSR	Logic shift left Logic shift right	Direct*	5	3	4	4	4.5
NEG	Negate	Direct*	5	3	4	4	4.5
ORA	Inclusive OR	Direct	3	—	3	—	—
ROL ROR	Rotate left Rotate right	Direct*	5	3	4	4	4.5
SBC	Subtract w/carry	Direct	3	—	3	—	—
SUB	Subtract	Direct	3	—	3	—	—
TST	Test for neg/zero	Direct	4	—	3	—	—

1) Instructions with asterisk (*) are read-modify-write instructions.
2) Data read is accessed up until to the end of the read cycle.
3) Data written at start of write cycle for port B and internal registers.
4) Data written halfway through write cycle for ports A and C only.

Understanding MMDS05 Timing

The timing given in **Figure 3** shows the relative signals used on the MMDS05 for the instruction decode. The LIR/RW signal from the MCU being used for emulation has a double pulse, but the rising edge of the first E clock generates the proper LIR signal. The start of each instruction opcode is, therefore, the rising edge of PH2 just as the LIR signal rises.

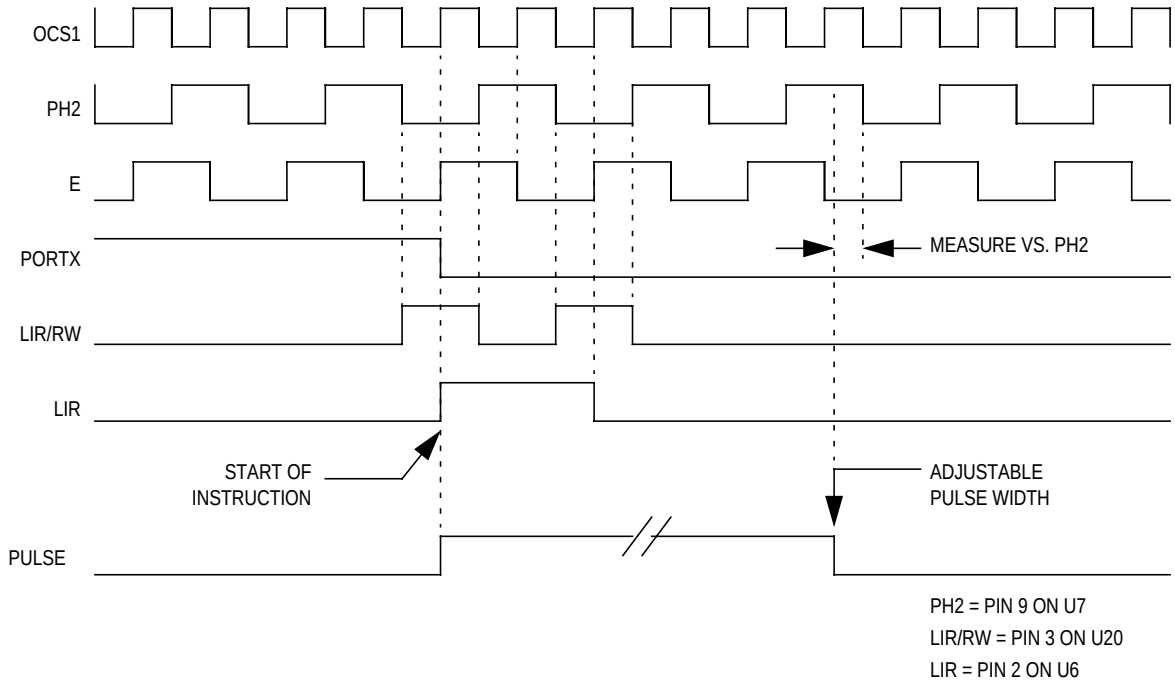

Figure 3. MMDS05 Timing

The actual setup used to measure exact read and write timing is shown in **Figure 4**. The steps are:

- The LIR signal is to drive two 74HC00 NAND gates in series (connected as inverters) from the LIR signal at pin 2 of U6 on the MMDS05 to the trigger input of the pulse generator. This buffers the trigger input which wants to see 50 inputs.
- The output of the pulse generator is then fed through two more 74HC00 NAND gates in series (also connected as inverters) to one input channel of a 4-channel oscilloscope.
- Another input of the oscilloscope is connected to the PH2 signal at pin 9 or U7 on the MMDS05.
- Another channel is connected to the trigger input of the pulse generator.

The software, *timing.asm*, used is given at the end of this application note.

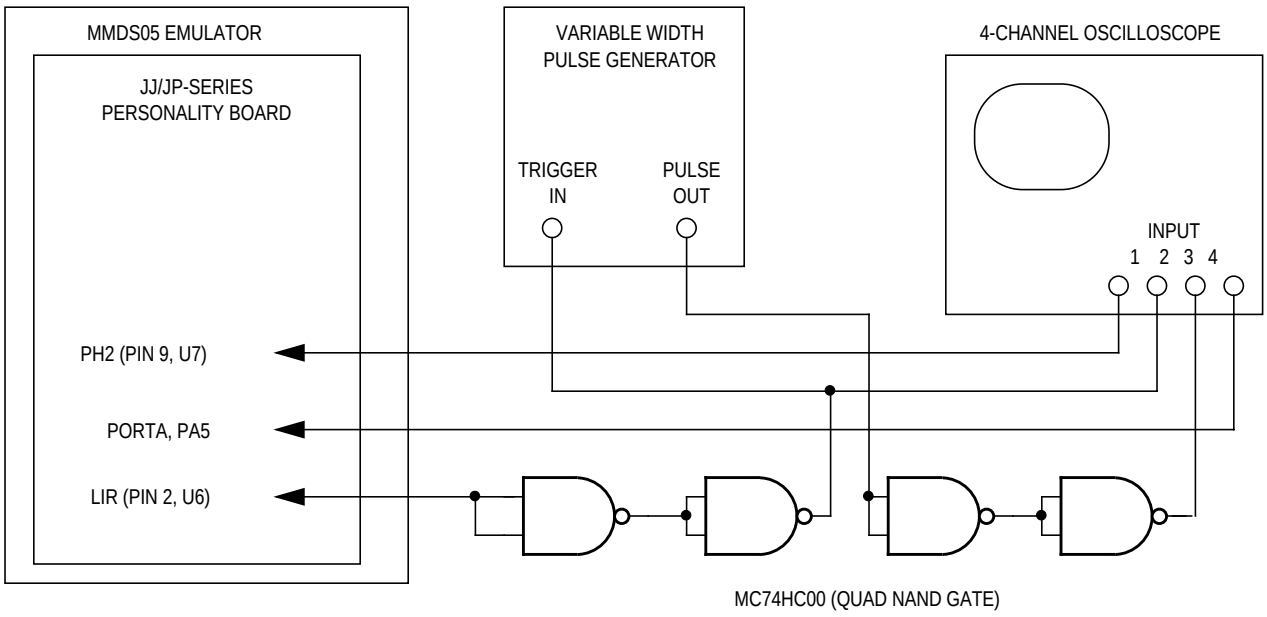

Figure 4. Timing Measurement Setup

For read timing, the software takes a reading on the desired input and then reflects the input state to the PA5 pin as an output. This PA5 output is then fed to another input on the oscilloscope. The pulse duration is adjusted to cause only a random change in the PA3 output during the expected read cycle of the instruction being used. The expected read cycle is counted from the rising edge of PH2 just as the LIR signal goes high (trigger input). The phase timing can be measured between the trigger input and the PH2 signal on pin 9 of U7 on the MMDS05.

The measured read timing for PORTA, PORTB, PORTC, and the comparator outputs is shown in **Figure 5**.

NOTE: *Even though the data begins to be read about halfway through the cycle, the actual state that is read and latched into the CPU occurs at the end of the read cycle.*

For write timing, the software simply toggles the state of the desired output. This output is then fed to another input on the oscilloscope. The change in the output with respect to the PH2 signal is measured during the expected write cycle of the instruction being used. The expected write cycle is counted from the rising edge of PH2 just as the LIR signal goes high (trigger input).

The measured write timing for PORTA, PORTB, PORTC, and the current source to PB0 is shown in **Figure 6**.

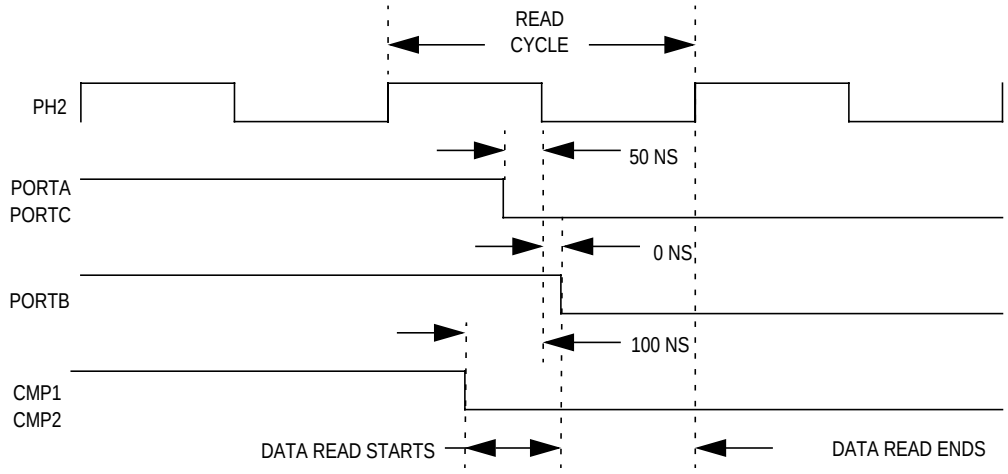

Figure 5. LDA Read Timing

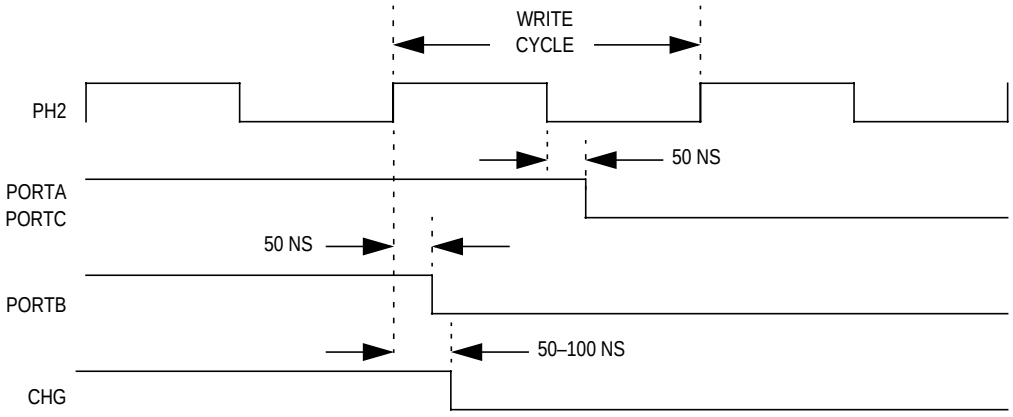

Figure 6. STA Write Timing

Referring to the Figure 5 and Figure 6, the timing for read and write operations is summarized in Table 3.

Table 3. Read/Write Cycle Timing

Operation	Source	Start vs. PH2 rise (ns)	Start vs. PH2 fall (ns)	End vs. next PH2 fall (ns)	Equiv. portion of PH2 cycle
READ	PORTA	—	–50	0	1.0
	PORTB	—	0	0	1.0
	PORTC	—	–50	0	1.0
	CMP1/CMP2	—	–100	0	1.0
Operation	**Destination**	**Start vs. PH2 rise (ns)**	**Start vs. PH2 fall (ns)**	**End vs. Next PH2 fall (ns)**	**Equiv. portion of PH2 cycle**
WRITE	PORTA	—	+50	0	0.5
	PORTB	+50	—	0	0.0
	PORTC	—	+50	0	0.5
	CHG	+75	—	0	0.0

Using the bus state analyzer in the MMDS05, the actual cycle used for read/write can be measured for all the various instruction types and their addressing modes as given in Table 1 and Table 2.

Timing Measurements on Other MC68HC05 MCUs

Similar timing measurements can be made on other members of the MC68HC05 Family of MCUs by using the technique shown in Figure 4. The user can find the appropriate PH2, LIR, and f_{osc} signals on the schematics supplied with each emulator personality module for the MMDS05.

Timing Measurement Software

The software included here was used to measure the timing of reads and writes on the JJ/JP series and can be adapted for other MCUs, as well.

timing.asm – Timing Test Setup Software

```
                    ****************************************************************
                    * TIMING Instruction Cycle Time Setup
                    *
                    * Designed for 28-pin JP6/7.
                    *
                     * Original:   Mark Shaw   12Aug97
                    * Revised:    Mark Shaw   06Sep97
                    ****************************************************************
                    ; MC68HC705JP7 definitions.
                    ;
ADDR CODE           ; Register addresses
0000                PORTA    equ     $00      ; port A data register
0001                PORTB    equ     $01      ; port B data register
0002                PORTC    equ     $02      ; port C data register
0003                AMUX     equ     $03      ; Analog Mux register
0004                DDRA     equ     $04      ; port A data direction register
0005                DDRB     equ     $05      ; port B data direction register
0006                DDRC     equ     $06      ; port C data direction register
000D                ISCR     equ     $0d      ; interrupt status/control register
0010                PDRA     equ     $10      ; port A & C pull-down register
0011                PDRB     equ     $11      ; port B pull-down register
001D                ACR      equ     $1d      ; analog subsystem control register
001E                ASR      equ     $1e      ; analog subsystem status register
                    ;--------------------------------------------------------------

                    ; Bit addresses
                    ;
                    : Port A (PORTA)
0005                PA5      equ     5        ; bit 5
0002                PA2      equ     2        ; bit 2
0000                PA0      equ     0        ; bit 0

                    ; Port C (PORTC)
0000                PC0      equ     0        ; bit 0

                    ; Interrupt status & control register (ISCR)
0006                OM2      equ     6        ; Oscillator mode select 2
0005                OM1      equ     5        ; Oscillator mode select 1

                    ;--------------------------------------------------------------
                    ;Memory addresses
                    ;
0020                RAM      equ     $0020    ; lowest RAM address
```

```
                    ;--------------------------------------------------------------
0800                        org     $0800
                    ;--------------------------------------------------------------
                    ; SETUP DEVICE
                    ;
0800 9C             BEGIN   rsp             ;Redundant reset of stack pointer
0801 9B                     sei             ;Redundant set of int. mask bit, I

0802 A660           SU1     lda     #$60    ;OM2 and OM1
0804 B70D                   sta     ISCR    ;Enable both the LPO and EPO

0806 A620           SU3     lda     #$20    ;OM1 only
0808 B70D                   sta     ISCR    ;Switch clock source to the EPO


                    ;--------------------------------------------------------------
                    ; SETUP PORTS
                    ;
080A A6FF           SU4     lda     #$FF
080C B710                   sta     PDRA    ;Inhibit port A & C pulldowns
080E B711                   sta     PDRB    ;Inhibit port B pulldowns

0810 B700           SU5     sta     PORTA   ;Set all port A pins, LEDs off
0812 B702                   sta     PORTC   ;Set all port C pins, LEDs off

0814 B704           SU6     sta     DDRA    ;Set up port A as outputs
0816 3F05                   clr     DDRB    ;Set up port B as inputs
0818 B706                   sta     DDRC    ;Set up port C as outputs

                    ;--------------------------------------------------------------
                    ; LOOP ON PORT TO FIND TIMING
                    ;
081A CC081D                 jmp     CHECK1  ;Jump to check of choice
                                            ;Change CHECK to start of
                                            ; of routine desired.

                                            ;Test read timing of ACR
081D 1A00           CHECK1  bset    PA5,PORTA ;Send out pip
081F 1B00                   bclr    PA5,PORTA


0821 A681                   lda     #$81    ;Set ICEN and CHG
0823 C7                     fcb     $C7
0824 00                     fcb     $00
0825 1D                     fcb     $1D
0826 A601                   lda     #$01    ;Clear CHG
0828 C7                     fcb     $C7
0829 00                     fcb     $00
082A 1D                     fcb     $1D
082B 20F0                   bra     CHECK1
```

```
082D A6FF          CHECK2  lda     #$FF       ;Testing write timing
082F B704                  sta     DDRA       ;Set ports as outputs
0831 B705                  sta     DDRB
0833 B706                  sta     DDRC

0835 A6FF          LOOP1   lda     #$FF       ;Set all port pins high
0837 B700                  sta     PORTA
0839 B701                  sta     PORTB
083B B702                  sta     PORTC

083D 4F                    clra               ;Set all port pins low
083E B700                  sta     PORTA
0840 B701                  sta     PORTB
0842 B702                  sta     PORTC

0844 20EF                  bra     LOOP1

                                               ;Test reads of comparator outputs
0846 1104          CHECK3   bclr    PA0,DDRA  ;Set PA0 as input
0848 1106                   bclr    PC0,DDRC  ;Set PC0 as input
084A A606                   lda     #$06
084C B71D                   sta     ACR       ;Turn on comps
084E A681                   lda     #$81
0850 B703                   sta     AMUX      ;Select PB1

0852 1A00          LOOP2    bset    PA5,PORTA ;Send out trigger
0854 1B00                   bclr    PA5,PORTA ;End trigger
0856 B61E                   lda     ASR       ;Fetch data
0858 A402                   and     #$02      ;Mask off PA0
085A 2704                   beq     LOW       ;If low, set PA2 low
085C 1400                   bset    PA2,PORTA ; else set PA2 high
085E 20F2                   bra     LOOP2

0860 1500          LOW      bclr    PA2,PORTA
0862 20EE                   bra     LOOP2

                   ;--------------------------------------------------------------
                   ; INTERRUPT TRAP
0864 20FE          ITRAP    bra     ITRAP

1FF2                        org     $1FF2
                   ;--------------------------------------------------------------
                   ; RESET AND INTERRUPT VECTORS
1FF2 0864                   fdb     ITRAP     ;Analog vector
1FF4 0866                   fdb     ITRAP     ;Serial vector
1FF6 0868                   fdb     ITRAP     ;Timer vector
1FF8 086A                   fdb     ITRAP     ;Core timer vector
1FFA 086C                   fdb     ITRAP     ;Ext IRQ vector
1FFC 086E                   fdb     ITRAP     ;SWI vector
1FFE 0800                   fdb     BEGIN     ;Reset vector
2000                        end
```

Application Note

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

freescale™ semiconductor