

## AN1740

## Applications Using the Analog Subsystem on MC68HC05JJ/JP Series Microcontrollers

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

### Introduction

---

The MC68HC05JJ/JP series of microcontrollers (MCUs) presents a unique combination of traditional digital peripherals with simple analog components which can be used to implement a variety of special functions.

These analog features include:

- A pair of voltage comparators
- Input channel multiplexer (MUX)
- A current source
- A temperature sensing diode
- Associated control bits and flags

The simple nature of these analog features requires very little silicon die area, but can provide functions normally seen in larger and more expensive MCUs. The designer has a great deal of flexibility in using these simple analog features; but with this flexibility comes the need to understand how these features can be combined and how they might

interact with each other. The intent of this application note is to present a range of uses for the MC68HC05JJ/JP analog features with details on how to set them up. It also points out some pitfalls the designer might encounter.

This application note covers these 12 topics:

- Analog subsystem overview
- Voltage comparators
- Current source/discharge
- Analog signal multiplexers
- Support circuitry
- Analog power-up considerations
- Analog-to-digital (A/D) conversion
- Pitfalls in using single slope A/D
- Manual A/D conversions
- Automated A/D conversions
- General A/D techniques
- Design check list

All members of the MC68HC05JJ/JP series of MCUs have the same analog subsystem and support timers, and they all have the same interface and control registers for these functions. Therefore, this application note covers all these members equally. The variations between series members are in memory type, memory size, number of I/O (input/output) pins, and personality EPROM. The designer should consult the individual MCU technical data book for specific information about the features on each member.

**NOTE:** *Examples of “typical” data are given in this application note so that the reader has an idea of the general behavior of the devices. These typical data are taken using a few samples of the devices and do not represent the full range of variation in the given parameter. For more information on the range of parameters, the reader should refer to the appropriate data sheet or consult the factory.*

## Analog Subsystem Overview

---

The block diagram of the analog subsystem is shown in [Figure 1](#). The analog subsystem is structured around three basic components:

- Two voltage comparators
- A constant current source

An assortment of channel multiplexers, control logic, and control/status registers are provided in order to use these basic building blocks in typical analog interfaces.

### Voltage Comparators

Each voltage comparator can compare voltages which are more than 1.5 volts below the  $V_{DD}$  supply voltage. The input offset is typically 10 millivolts and typical response time is less than a microsecond. Both comparators have identical performance and differ only in how they are connected to external pins. Each comparator can have its internal circuitry “flipped” so that any input offset can be referenced to either input. The value of this is included in the discussion of A/D techniques. Each comparator also can be powered down to conserve supply current when its voltage comparison function is not needed.

### Constant Current Source

A constant current source is provided to an external pin and sources typically 100  $\mu$ A as long as the voltage on the pin is more than 1.5 volts below the  $V_{DD}$  supply voltage. The current being sourced can be controlled by a bit that also turns off its internal  $V_{DD}$  supply current when it is not being used. The primary purpose of the current source is to construct an integrating A/D using voltage comparator 2. But, this current source can be used for other purposes if an external device can utilize the nominal 100- $\mu$ A current.

### Discharge Device

A discharge device is also attached to the constant current source output. Either the current source is ON, the discharge device is ON, or both are OFF. This allows control of an external charge storage device,

## Application Note

such as a capacitor, to be attached to PBO and used as an integrator which can be reset.

**Channel Multiplexer**

Voltage comparator 2 can be interconnected to external pins and internal sources using a 6-channel multiplexer.

**Input Divider**

The channel selection multiplexer can feed a voltage divider which allows the input signals to the channels to be higher than the common mode range of voltage comparator 2.

**Sample and Hold**

A provision at the input of comparator 2 is made for a sample and hold on an internal capacitor. The source for sample and hold can be the output of either the channel multiplexers or the input divider.

**On-Chip Temperature Sensing Diode**

The temperature of the device can be inferred by measuring the voltage on the internal diode. This voltage can be measured using the simple A/D convertor techniques discussed later.

**Control/Status Registers**

To use these basic building blocks in typical analog interfaces, there is an assortment of channel multiplexers, control logic, and three registers containing 20 control bits and four status bits related to the analog subsystem. All the registers relating to the analog subsystem are shown in **Figure 2**.

These registers are:

- Analog MUX register (AMUX) at \$0003
- Analog control register (ACR) at \$001D
- Analog status register (ASR) at \$001E

**NOTE:** *Some analog features are enabled or disabled by an OPT bit. This OPT bit is in the mask option register (MOR) in the EPROM versions or is specified as a mask option on the ROM versions.*

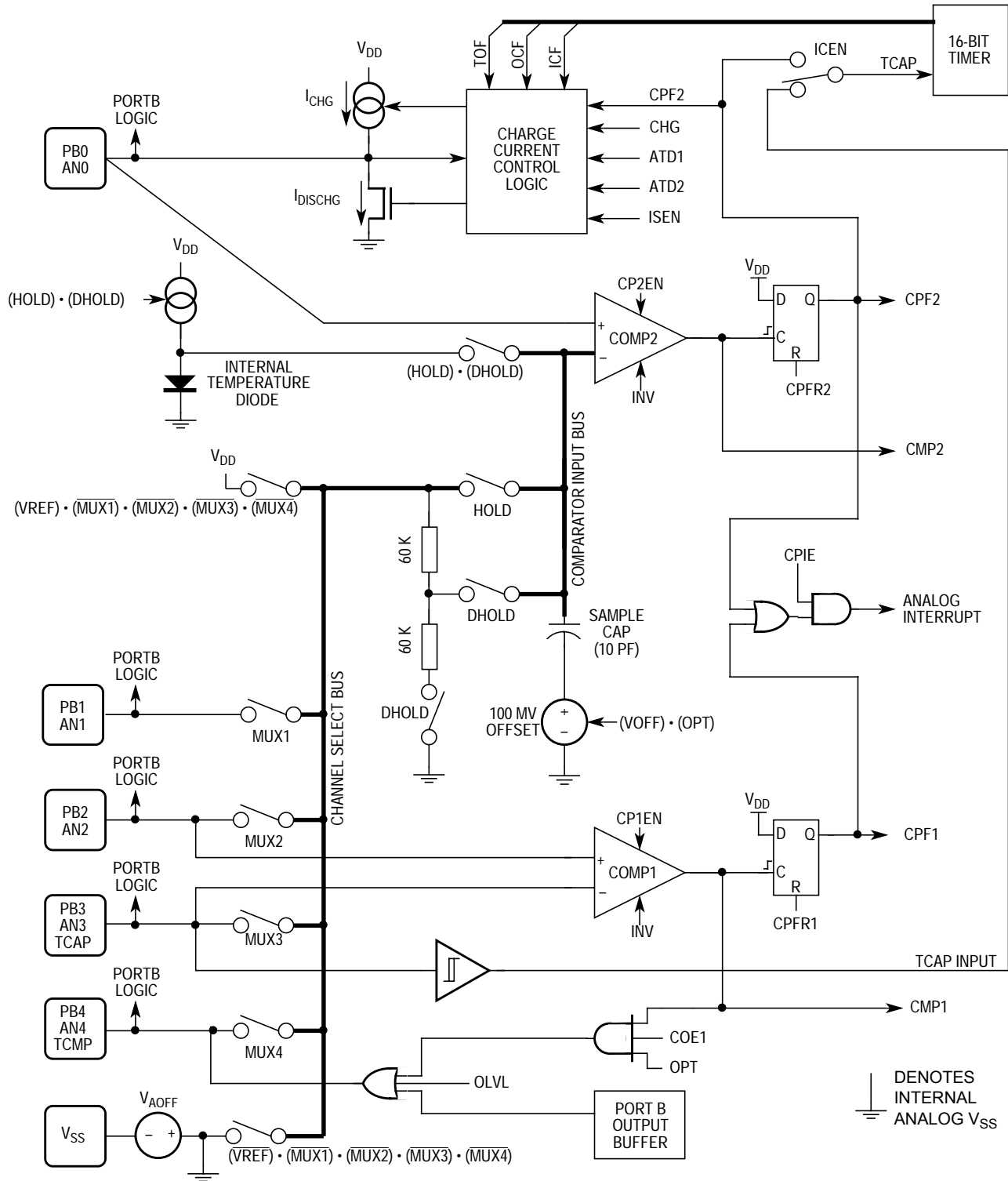


Figure 1. Analog Subsystem Block Diagram

**Application Note**

Addr.	Register		Bit 7	6	5	4	3	2	1	Bit 0
\$0001	Port B data, PORTB	Read: Write:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
\$0003	Analog MUX register, AMUX	Read: Write:	HOLD	DHOLD	INV	VREF	MUX4	MUX3	MUX2	MUX1
\$0005	Port B data direction, DDRB	Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
\$0011	Port B pulldown, PDRB	Read: Write:								
\$0012	Timer control, TCR	Read: Write:	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
\$0013	Timer status, TSR	Read: Write:	ICF	OCF	TOF	0	0	0	0	0
\$0014	Input capture MSB, ICRH	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
\$0015	Input capture LSB, ICRL	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
\$0016	Output compare MSB, OCRH	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
\$0017	Output compare LSB, OCRL	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
\$001D	Analog control, ACR	Read: Write:	CHG	ATD2	ATD1	ICEN	CPIE	CP2EN	CP1EN	ISEN
\$001E	Analog status, ASR	Read: Write:	CPF2	CPF1	0	0	VOFF*	COE1*	CMP2	CMP1
				CPFR2	CPFR1					



= Unimplemented

\* Controlled by mask option.

**Figure 2. Analog Subsystem Related Registers**

## Voltage Comparators

---

The voltage comparators are a basic design, but allow software compensation for the offset voltage. **Table 1** shows the main features of both voltage comparators. The common-mode voltage range is sufficient for most low-level signal sources, but may be a limiting factor for some applications where the input source must swing from  $V_{SS}$  to  $V_{DD}$ .

**Table 1. Typical Voltage Comparator Features**

Characteristic	Symbol	Typ	Units
Common mode input range	$V_{CMR}$	$V_{DD}-1.5$	V
Input offset	$V_{IO}$	10	mV
Input impedance (with input pulldowns OFF) Direct, either comparator With divider connected to comparator 2	$Z_{IN}$ $Z_{IN}$	1000 120	k $\Omega$ k $\Omega$
Response time Output Switching (10 mV overdrive) Power up delay (analog bias ON) Power up delay (analog bias OFF)	$t_{CPROP}$ $t_{CDELAY}$ $t_{CDELAY}$	1 2 5	$\mu$ s $\mu$ s $\mu$ s
Supply current Comparator only Bias source for both comparators	$I_{DDC}$ $I_{DDC}$	100 65	$\mu$ A $\mu$ A

### Comparator 1

Comparator 1 is a basic voltage comparator with its positive and negative inputs tied permanently to the PB2 and PB3 pins respectively. Its output can be monitored by both dynamic output and static flag bits in the ASR (analog status register) located at address \$001E. The dynamic output bit, CMP1, becomes a logical 1 whenever the comparator's positive input is above its negative input and it follows the output of the comparator regardless of its prior conditions. The static flag bit, CPF1, becomes set whenever there is a rising output from the comparator and remains set until cleared by writing a logical 1 to the reset bit, CPFR1. Therefore, the CPF1 bit can be used to capture events when the comparator's positive input was above its negative input. Comparator 1, therefore, permits software subroutines to either poll the

CMP1 or CPF1 bits or use an analog interrupt that is generated when the CPF1 bit is set.

Comparator 1 also has the provision to connect its output to the PB4 pin by setting the COE1 bit in the ASR. This makes this comparator useful for directly driving some external function. (The use of the COE1 bit is enabled by the OPT mask option.)

The connections to PB2, PB3, and PB4 mean that the user must decide the dominant use of these pins. While the inputs to comparator 1 will not affect the levels applied to PB2 and PB3 (even if above the common-mode range of comparator 2), the output of comparator 1 can be OR'ed with any internal source driving PB4. Comparator 1 can be used in a variety of configurations which are described in the following paragraphs and shown in [Figure 3](#).

**Case 1** uses comparator 1 as a simple function to detect when the voltage on PB2 rises above PB3. In this case, port B would be configured with PB2 and PB3 as inputs. Normally, the user would deactivate the input pulldowns on PB2 or PB3, but there is no internal reason that they could not remain active if the external voltage sources can supply up to 300  $\mu$ A. A software subroutine can then be initiated by polling the state of the CMP1 output bit or the CPF1 flag bit or by an analog interrupt caused by CPF1 being set.

**Case 2** uses comparator 1 solely for an outside function. In this case, the user should configure the PB2 and PB3 pins to be inputs with their pulldowns deactivated and the PB4 pin as an output. The COE1 bit should be set with both the OLVL and PB4 data bit cleared.

**Case 3** uses comparator 1 as an additional detection means for another MCU pin. As in case 2, the PB2 and PB3 pins should be configured as inputs with their pulldowns deactivated and the PB4 pin as an output. The COE1 bit should be set with both the OLVL and PB4 data bit cleared. The comparator's output can then be wrapped around from the PB4 pin to the  $\overline{\text{RESET}}$ ,  $\overline{\text{IRQ}}$ , or a port A pin which has IRQ capability (PA0:PA3). This essentially creates an adjustable threshold for resetting the MCU or causing an IRQ interrupt. Possible uses would be low battery detection, level alarms, and low-voltage reset. If the comparator 1 output drives the IRQ function, the additional IRQ source can be made



a lower priority than other IRQ sources by polling the CMP1 or CPF1 bits in the IRQ subroutine. If CPF1 is set, the IRQ subroutine could branch within the IRQ subroutine or cause the IRQ subroutine to be exited and, therefore, allow an analog interrupt to occur. In either case, the subroutine called should clear the CPF1 bit so that future detection is possible.

**Case 4** uses comparator 1 as a monitor, but also uses the digital inputs on PB2 and PB3 as a crude threshold detector. In this case, the software can determine whether the comparator is checking voltages above or below the digital threshold of port B. The digital threshold is typically 40% to 60% of the  $V_{DD}$  supply voltage. In this case, it is important that the PB2 and PB3 pins are configured as inputs with their pulldowns disabled.

**Case 5** uses comparator 1 as a monitor, but it also uses the digital output on PB4 to alter the trip point for the comparator. As shown in **Figure 3**, the PB4 output buffer could shunt the external reference divider causing the comparator to set when the parallel combination of R1 and R3 is connected to  $V_{DD}$  and then reset when the parallel combination of R2 and R3 is connected to  $V_{SS}$ . Careful review of the port B description also will show that the shunting of the external reference divider can be done by the OR combination of the PB4, TCMP, or CMP1 functions so that the control can be based on software control, timer control, or direct feedback from comparator 1. In this case, it is important that the PB2 and PB3 pins are configured as inputs with their pulldowns disabled.

Application Note

Freescale Semiconductor, Inc.

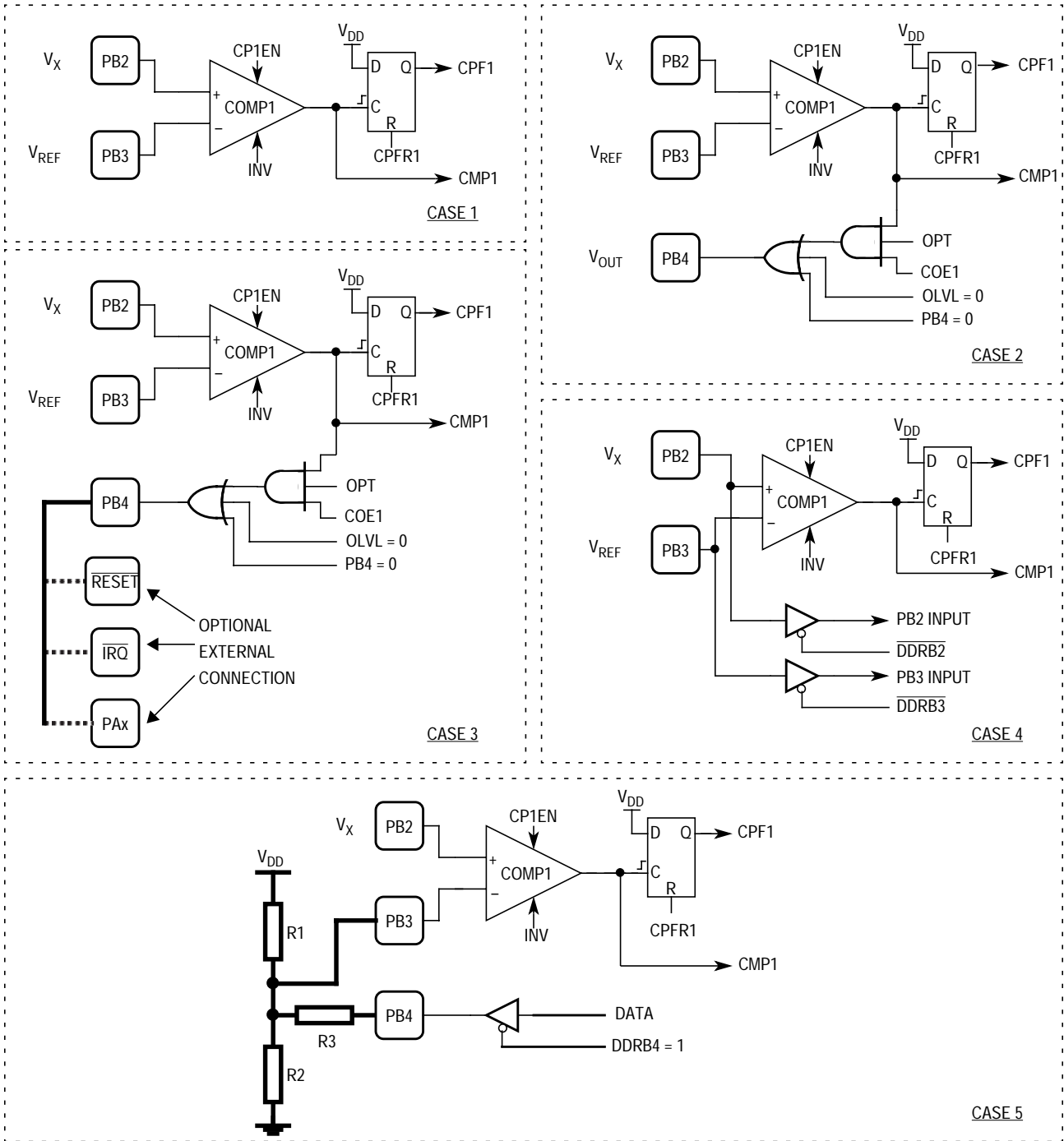


Figure 3. Comparator 1 Configurations

## Comparator 2

Comparator 2 is a basic voltage comparator with its positive input tied permanently to the PB0 pin and its negative input being tied to a number of internal or external functions. The negative input of comparator 2 can be switched to various sources, but in all cases the internal sample capacitor of approximately 10 pF will remain connected from the negative input to  $V_{SS}$ . Its output can be monitored by the dynamic output and static flag bits in the ASR located at address \$001E. The dynamic output bit, CMP2, follows the output of the comparator regardless of its prior conditions and becomes a logical 1 whenever the comparator's positive input is above its negative input. The static flag bit, CPF2, becomes set whenever there is a rising output from the comparator and remains set until cleared by writing a logical 1 to the reset bit, CPF2R. Therefore, the CPF2 bit can be used to capture events where the comparator's positive input rose above its negative input.

Not only does comparator 2 allow software subroutines polling the CMP2 or CPF2 bits or an analog interrupt caused by CPF2 being set, the CPF2 flag bit also can trip the input capture function of the 16-bit timer if the ICEN bit is set in the ACR at location \$001D.

Comparator 2 has limited connection capabilities versus comparator 1. The primary purpose of comparator 2 is to construct a multiple channel integrating A/D convertor using the internal channel MUX, internal references, input divider, and sample and hold. Comparator 2 can be used as a simple comparator as shown in [Figure 4](#). These uses are similar to cases 1, 4, and 5 for comparator 1 shown in [Figure 3](#). However, due to the internal resources, comparator 2 has several unique cases of its own. These additional cases are shown in [Figure 5](#) and are described in the following paragraphs.

**Case 6** uses comparator 2 to monitor an outside function on PB0 with respect to the divided input on PB1, PB2, PB3, or PB4. In this case, the user should configure the PB1:PB4 input selected to be an input with its pulldown deactivated. The divider nominally divides the input by 2 which allows the selected PB1:PB4 pin to be tied to a voltage level up to  $V_{DD}$ . To activate the divider, the DHOLD MUX must be enabled.

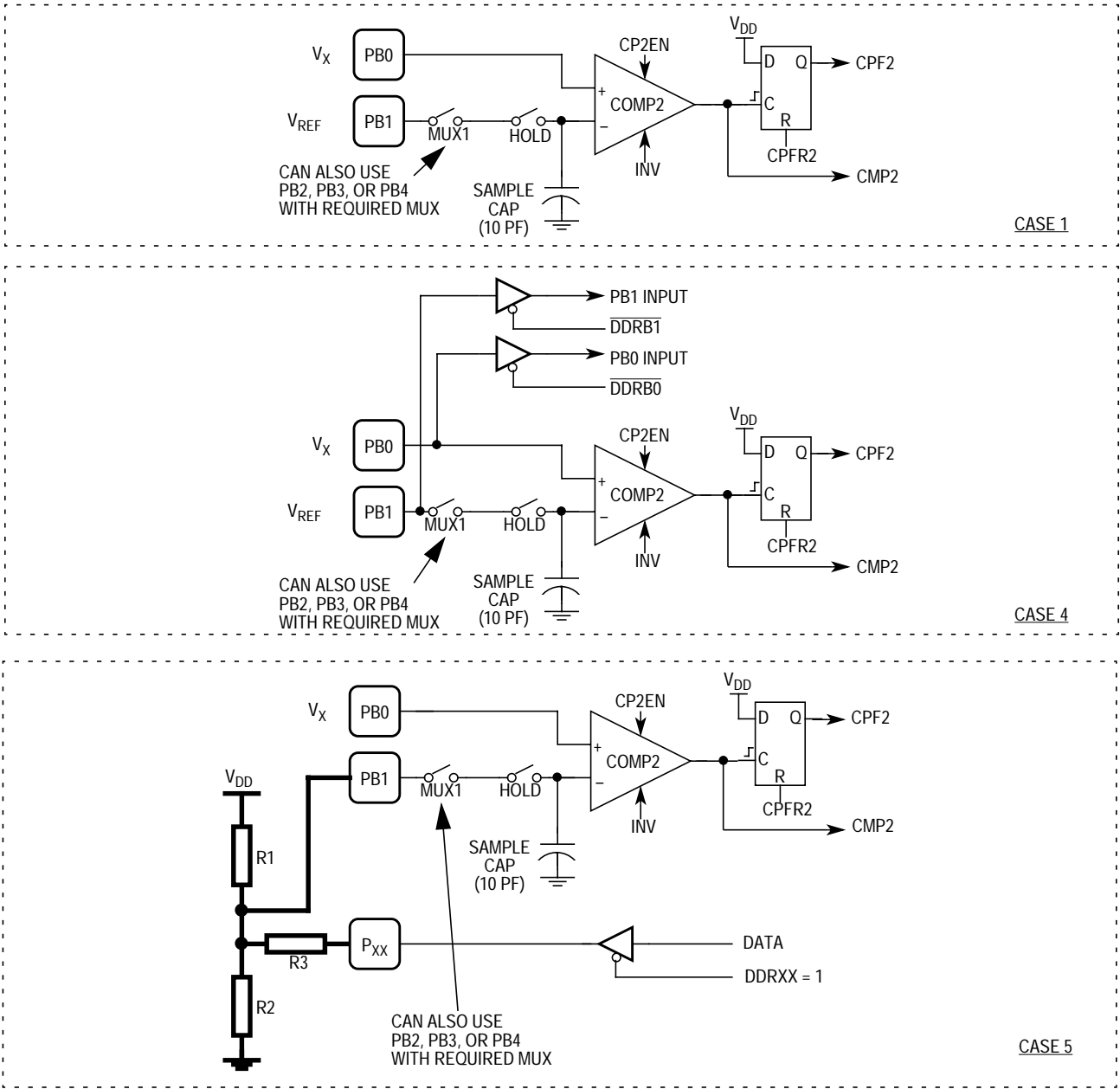


Figure 4. Comparator 2 Configurations

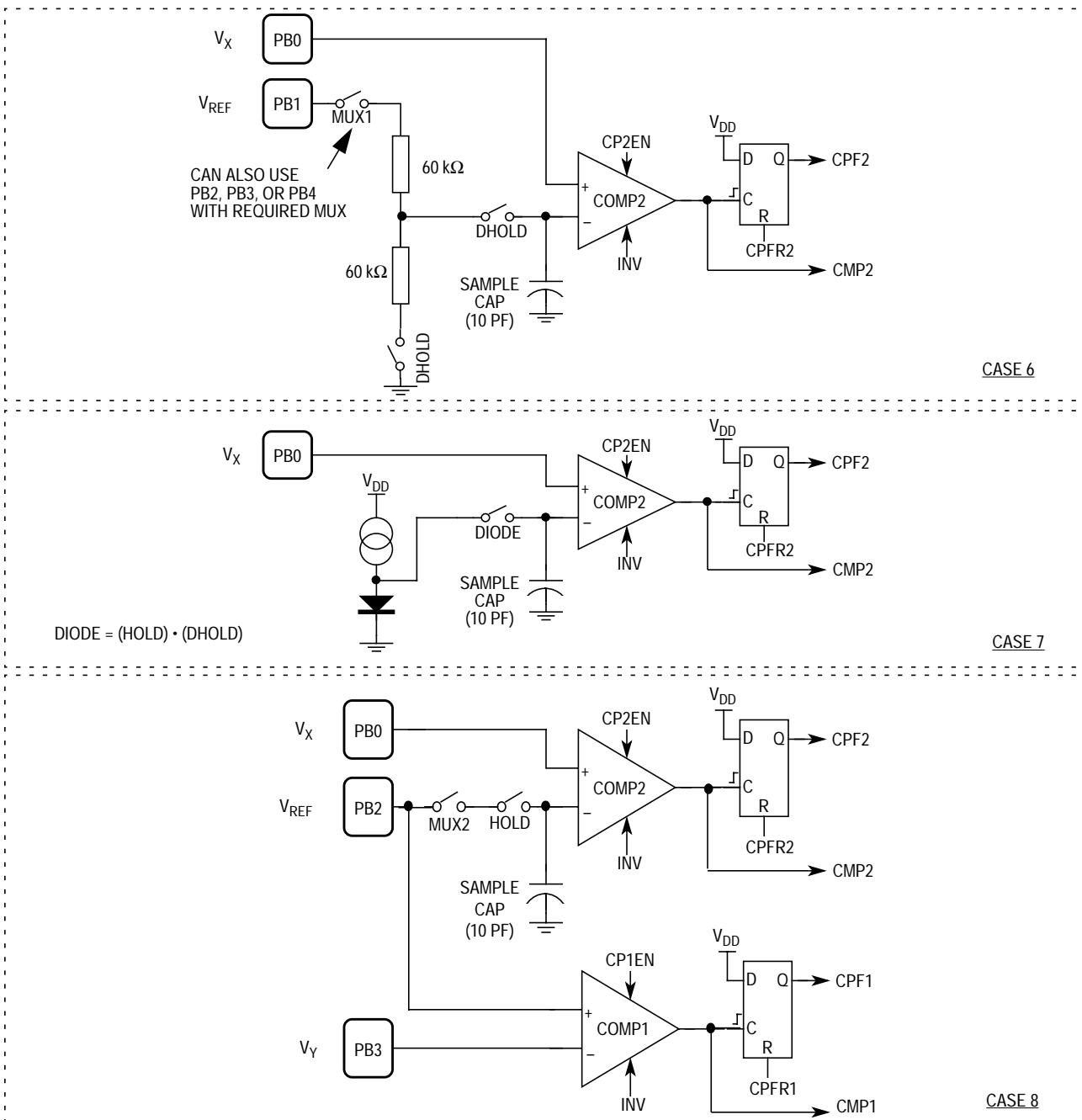


Figure 5. Additional Comparator 2 Configurations

**Case 7** uses comparator 2 as a simple function to detect when the voltage on PB0 rises above the internal temperature sensing diode. In this case, port B would be configured with PB0 as an input. Normally, the user would deactivate the input pulldown on PB0, but there is no internal reason that it could not remain active if the external voltage source to PB0 can supply up to 300  $\mu$ A of current to the pulldown.

**Case 8** shows how both comparators can be tied to a common signal by using the internal channel MUX and leaving the HOLD MUX enabled. Notice that the negative input of comparator 2 can be tied to either the positive or negative input of comparator 1 by MUX2 or MUX3. If the positive input of comparator 2 needs to be tied to either input of comparator 1, then an external connection can be used.

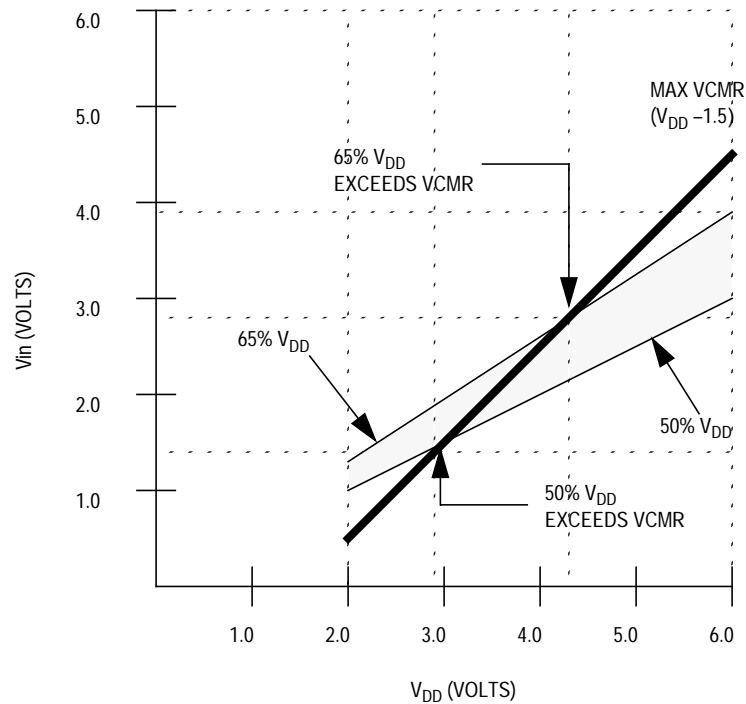
#### Comparator Common Mode Range

The comparators can determine only which input is higher as long as at least one input stays within the common mode range from  $V_{SS}$  to  $V_{DD} - 1.5$  volts. If both inputs should fall outside this range, then the comparator's output will be undefined. It is important to constrain at least one input into the common mode range for reliable operation. Taking the input above  $V_{DD} - 1.5$  volts is only recommended when using the divided input to comparator 2.

Care must be taken when using sources ratiometric to  $V_{DD}$  to ensure that the common mode limits are not exceeded as the  $V_{DD}$  level changes. When the supply voltage decreases, the common mode range decreases at a more rapid rate than the percentage drop in  $V_{DD}$ . For example, consider an input source that only varies between 50% and 65% of  $V_{DD}$ . As shown in **Figure 6**, when the  $V_{DD}$  level is at 5.0 volts, the input will stay between 2.50 to 3.25 volts which are both within the common mode limits. However, the common mode range is defined by a fixed voltage drop below  $V_{DD}$ . As  $V_{DD}$  falls to 4.29 volts, the upper range of the ratiometric input is now at 2.79 volts, which is right at the limit of the common mode range. As  $V_{DD}$  falls farther, the input can become greater than  $V_{DD} - 1.5$  volts. As  $V_{DD}$  approaches the minimum operating limit of 2.7 volts, the input source range is completely above the common mode range.

**NOTE:** *Be sure to calculate the common mode range over the range of voltage levels expected for  $V_{DD}$  and always keep one input to the voltage comparators within the common mode range.*

In practice, the common mode range can extend below  $V_{SS}$  until the internal input protection diode starts to conduct at about  $-0.3$  volts (at  $125\text{ }^{\circ}\text{C}$ ). If the input is taken further negative, the diode conduction can become excessive which can seriously affect analog functions. In a like manner, the inputs can be taken up to  $0.3$  volts above  $V_{DD}$  before the upper input protection diode starts to conduct.



**Figure 6. Ratiometric Sources vs. Common Mode Range**

**Comparator Input Offset Voltage**

The voltage comparator design used in the MC68HC05JJ/JP series of MCUs was selected for its simplicity and small silicon area on the device. Basic comparators like this can have input voltage offsets of up to 15 millivolts. This amount of offset can be significant, especially when monitoring signals near  $V_{SS}$ . If one of the comparator inputs must go below  $V_{SS}$  to overcome the  $V_{IO}$ , then the comparator’s output will not change state and the comparator will look “stuck” even though both

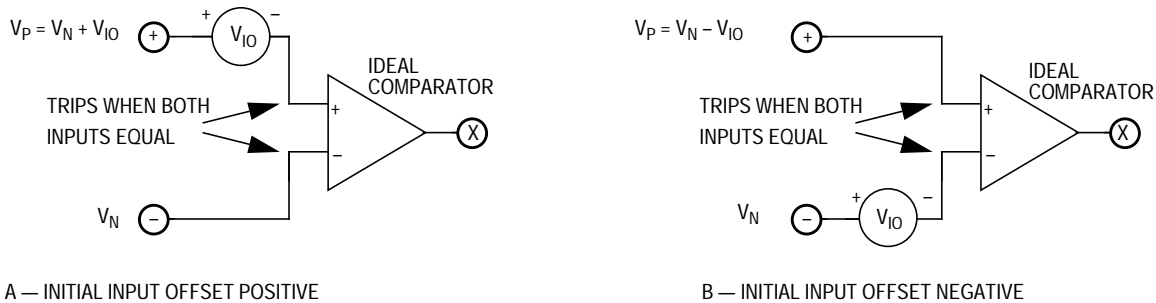
inputs may be within the common mode range above  $V_{SS}$ . The input offset voltage of a comparator can be modeled as a small internal voltage source in series with either input of a voltage comparator as shown in **Figure 7**.

In case A, the small  $V_{IO}$  source is in series with the positive input of the comparator. If both inputs to the comparator are to be equal, then the applied positive input must rise an additional amount equal to  $V_{IO}$ . Thus, the comparator trip condition occurs when:

$$V_P = V_N + V_{IO}$$

In case B, the small  $V_{IO}$  source is in series with the negative input of the comparator. If both inputs to the comparator are to be equal, then the applied positive input must fall an additional amount equal to  $V_{IO}$ . Thus, the comparator trip condition occurs when:

$$V_P = V_N - V_{IO}$$



IN BOTH CASES, THE OUTPUT (X) RISES WHEN THE (+) INPUT RISES ABOVE THE (-) INPUT.

**Figure 7. Comparator Input Offset Model**

**Comparator Inversion**

If the input offset can be moved between the inputs, then the sign of its contribution to  $V_P$  can effectively change. Since the input offset is actually the result of silicon device sizes and their matching, it cannot actually “move” between inputs. But the sense of the inputs can be redefined by interchanging what the inputs to the comparator are called while also inverting the comparator’s output sense as shown in **Figure 8**. In these cases, the actual offset stays with the same internal

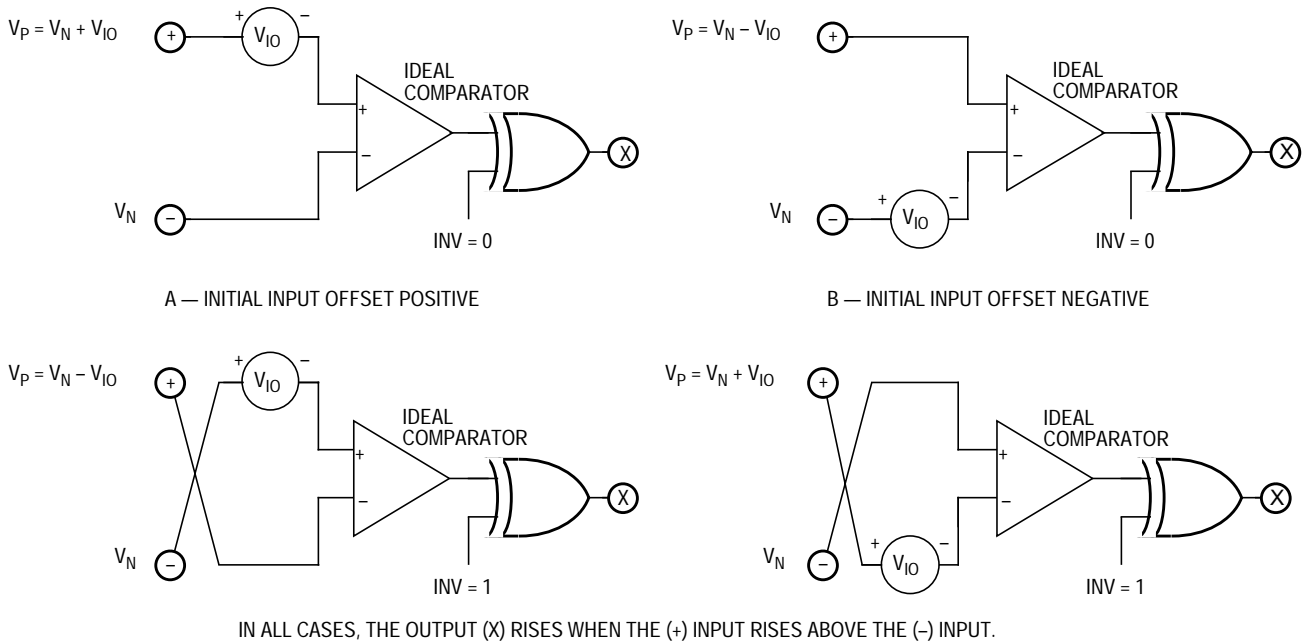


comparator input, but results in the addition or subtraction from the same external input to the comparator. This inversion action is performed by the INV bit in the AMUX register.

The INV bit, therefore, can be used to check for an input condition where the difference between the inputs is less than the offset voltage or for the case where one input is within the offset voltage of  $V_{SS}$ . For example, in case B in **Figure 8** where  $V_N$  is 5 mV above  $V_{SS}$  and  $V_{IO}$  is 10 mV with respect to the internal negative input of the comparator, then:

$$V_P = V_N - V_{IO} = 5 \text{ mV} - 10 \text{ mV} = -5 \text{ mV}$$

Therefore, the comparator output will not change states, since the  $V_P$  must be at least 5 mV below  $V_{SS}$ .



**Figure 8. Inverted Comparator Offset Model**

However, if we invert, the sense of the comparator is inverted using the INV bit, then  $V_N$  will still be at 5 mV, but it is now connected to the internal positive input of the comparator. The 10-mV input offset is still with respect to the internal negative input, but now is “moved” to the external  $V_P$  so that:

$$V_P = V_N + V_{IO} = 5 \text{ mV} + 10 \text{ mV} = + 15 \text{ mV}$$

Therefore, the comparator will now change states if the voltage on  $V_P$  is near 15 mV.

In this discussion, the offset voltage has been described with respect to the negative input of the comparator, but in practice this offset may be with respect to either input of the comparator. Therefore, the resulting output states for each state of the INV bit gives the user an indication of the relative voltages on  $V_P$  and  $V_N$  as given in [Table 2](#).

The two conditions where the comparator outputs change states based on the state of the INV bit can be further analyzed to show that each of these conditions is unique to a given polarity of the input offset voltage.

**Table 2. Results Using INV Bit on Comparators**

Output state of comparators		Polarity of input offset (see <a href="#">Figure 7</a> )
INV = 0	INV = 1	
0	1	Case A, positive, $V_P = V_N + V_{IO}$
1	0	Case B, negative, $V_P = V_N - V_{IO}$

The user, therefore, can determine the polarity of the input offset voltage for any particular comparator by simply connecting both of its inputs to the same voltage and then checking the comparator output state for each state of the INV bit versus [Table 2](#). The inputs can both be tied to  $V_{SS}$  by using the associated port B pins as outputs, if the inputs are connected to a source impedance greater than 2 k $\Omega$ . In this case, the user should first clear the associated port B data bit and then set the associated data direction bit. Another approach to tie together the inputs of comparator 1 is to set the MUX2 and MUX3 bits and clear the HOLD, DHOLD, MUX1, MUX4, and VREF bits in the AMUX register. This will connect the PB2 and PB3 pins by connecting them both to the channel multiplexer, which also has been isolated from other sources.

**NOTE:** *When operating with one comparator input near  $V_{SS}$ , it is recommended that the input always be tested with the INV bit both set and clear. If the results are different, then the inputs are both within  $V_{IO}$  of  $V_{SS}$ .*

It is impossible to control the outputs of the comparators when changing the state of the INV bit since it interchanges the inputs and inverts the output of both comparators at the same time. Therefore, the comparator

flags, CPF1 and CPF2 in the ASR, may get set during any change in the INV bit.

The user should use caution also when changing the state of the INV bit if either flag bit is enabled to trigger a timer input capture or analog interrupt. Any change to the INV bit should be done with these interrupts disabled or during a time when an interrupt is not expected before the flag bits can be cleared.

**NOTE:** *Changes to the INV, CP1EN, CP2EN, and ICEN bits can cause the comparator static flags to get set. Always clear the CPF1 and CPF2 static flag bits after changing any configuration of the comparators by writing a logical 1 to both the CPFR1 and CPFR2 bits in the ASR.*

#### Comparator Input Impedance

The comparators themselves have a very high input impedance, but several other circuits connected to their inputs will lower this ideal impedance. These currents or equivalent impedances are:

1. Input protection diodes for ESD which may have leakage currents up to  $\pm 2 \mu\text{A}$  (equivalent of  $2.5 \text{ M}\Omega$  at 5 volts)
2. Parasitic capacitances in the input and multiplexer circuitry will cause the input impedance to decrease with frequency as shown in [Figure 9](#).
3. The input divider of 2:1 has an impedance of  $120 \text{ k}\Omega$  at DC, but decreases at high frequencies due to the sample capacitor as shown in [Figure 9](#).
4. Software-controlled input pulldown devices which can sink up to  $280 \mu\text{A}$  (equivalent of  $18 \text{ k}\Omega$  at 5 volts)
5. More than one input multiplexer being on at the same time and connecting the comparator input to other low impedances
6. Output devices on port B pins being active

The ESD protection (1) and the parasitic capacitances (2) are always present and will be the limiting factor on high input impedance. The divider (3) only affects the negative input of comparator 2 and can be removed from the input by not using the DHOLD bit in the AMUX register. The last three sources (4, 5, 6) all can be controlled by the

Application Note

user's software code, but they can be a problem if they are overlooked in the application.

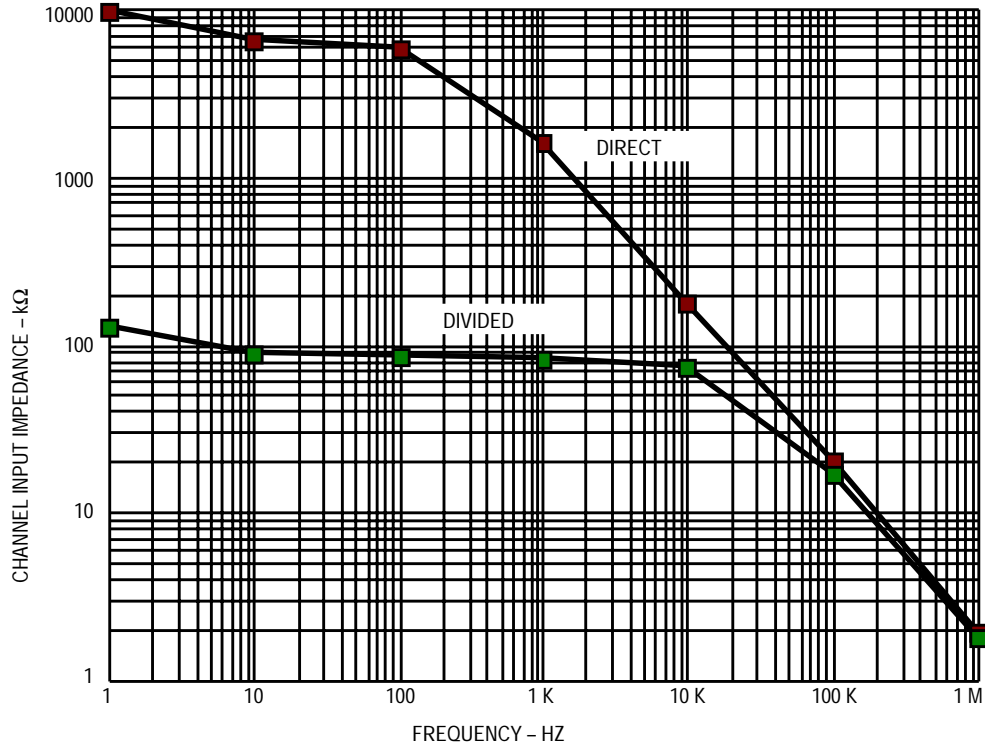


Figure 9. Typical Channel AC Input Impedance

**NOTE:** The most common cause of unwanted input current or low input impedance is a failure to turn off the input pulldown devices on port B. This is accomplished by writing logical 1s to the desired PDIB bits in the PDRB at location \$0011.

Freescale Semiconductor, Inc.

**Comparator Speed**

The comparators have good speed and can power up or respond to their inputs changing within 2 microseconds. However, powering up a comparator when neither comparator nor the current source was previously active can result in a longer delay in stabilization for both the analog bias source and the comparator. It is, therefore, recommended that at least 10 microseconds of delay be provided by the software to allow the comparator to stabilize following a power up before trying to read or reset its output flag bits. The simplest way to add a time delay would be a series of NOP instructions (at  $f_{OSC}$  of 4.2 MHz, each NOP takes 1 microsecond) or to insert other code which does not rely on the state of the comparator output flags.

**Comparator Power Consumption**

Each comparator typically draws 100  $\mu A$  of  $I_{DD}$  current when it is powered up by its appropriate CP1EN or CP2EN bit in the ACR. In addition, when either or both comparators are powered up, an analog bias source is also powered up which draws typically 65  $\mu A$ . This means that a single comparator will draw about 165  $\mu A$  and both comparators will draw about 265  $\mu A$ .

**Comparator Output Sampling**

There are two means to determine if a comparator has changed states:

- The static flags, CPF1 and CPF2
- The dynamic outputs, CMP1 and CMP2

The static flags are set whenever there is a rising edge on the output of the respective comparator, and they remain latched until cleared by writing a logical 1 to the respective CPFR1 and CPFR2 bits or by a reset of the device. These flags are useful for capturing events while the CPU is doing some other task. These flags also can generate an interrupt using the analog interrupt. In the case of comparator 2, a timer input capture interrupt also can be generated. As mentioned above, the state of the CPF1 and CPF2 flag bits should be ignored after changing the state of the INV bit until both flags have been reset using the CPFR1 and CPFR2 reset bits.

The dynamic output bits merely follow the state of the output of each comparator. There is no timing synchronization in passing the state of the CMP1 and CMP2 bits to the ASR. The dynamic output bits are useful

in testing the state of the comparator following the initiation of a static flag or simply to check that the comparator state is steady. Since there is no latching of the outputs, the user should be cautious about using the CMP1 or CMP2 bits when inputs to the comparator are changing at frequencies greater than the rate at which the software is sampling the ASR.

**Shared Functions on PB4**

One pin of port B, PB4, can be shared with the output of comparator 1, the output compare of the 16-bit timer, and the PB4 port I/O buffer itself. All of these functions are OR'ed together to drive the pin when it is configured as an output by setting the DDRB4 bit in the port B data direction register at location \$0005.

When the PB4 pin is configured as an output, the pulldown device is automatically disabled and the output compare and port B data register are OR'ed together. The output of comparator 1 can be added to the OR function if the COE1 bit is set and the additional analog options are selected by the OPT mask option.

Once the OR function is active, the user must disable the functions that are not wanted by clearing the correct bits as given in [Table 3](#).

**Table 3. Logical OR Function on PB4**

Desired output driving function on PB4	Register bit requirements				
	Port B data bit, PB4	16-bit timer output compare level, OLVL	OPT mask option, MOR or ROM	Comparator 1 output enable, COE1	Comparator 1 output, CMP1
Port B data register	PB4 data bit state	0	0	X	X
			1	0	X
				1	0
16-bit timer output compare	0	Timer OLVL state	0	X	X
			1	0	X
				1	0
Comparator 1 output	0	0	1	1	CMP state

**Comparator  
Output Analog  
Interrupt**

The outputs of the comparators can be monitored also by using the analog interrupt. When using interrupt-driven detection of the transitions, the main consideration is the latency of entering the interrupt service routine and its later return to the background software. The CPU will always complete the instruction currently in process before beginning an interrupt service routine. The range of instruction times varies from two to 11 CPU bus cycles.

The three longest instructions are the MUL (11 cycles), the SWI (10 cycles), and the RTI (nine cycles). While the user can avoid using either the MUL or SWI instruction, the RTI is a necessary instruction if interrupts are being considered.

Once the interrupt is acknowledged, there is a delay of 10 cycles to stack the CPU state before executing the first instruction in the interrupt service routine. This means it takes from 12 to 21 cycles to acknowledge the interrupt and access its service routine. Once the service routine is completed, there will be nine cycles to execute its final RTI instruction and 10 more cycles to unstack the CPU state before the interrupt is complete and the CPU is ready to begin processing another one.

From this discussion, at least 46 CPU bus cycles are required to simply acknowledge and clear the CPF1 or CPF2 static flags as follows:

- 21 cycles to acknowledge; max instruction plus stacking
- Six cycles to clear the static flag (BSET)
- 19 cycles to return; RTI plus unstacking

This does include any software time to actually perform a task. Therefore, analog interrupts should not be used to detect pulses separated by less than about 60 CPU bus cycles. On the other hand, software polling can test and clear the CPF1 or CPF2 static flags within 11 CPU bus cycles as follows:

- Five cycles to test for the static flag being set (BRSET)
- Six cycles to clear the static flag (BSET)

Software polling should not be used for pulses separated by less than 30 CPU bus cycles.

**NOTE:** Avoid using both the input capture and the analog interrupt at the same time for the comparator trip detection. In this case, the ICF always wins priority and the analog interrupt essentially will be ignored.

## Current Source/Discharge

---

The analog system's other significant feature is the constant current source with discharge device that is connected directly to the PB0 pin. As given in [Table 4](#), the current source and the discharge device can be operated in three modes which are dependent on the states of the ISEN and CHG bits in the ACR located at \$001E. ISEN controls whether the function is active and the CHG bit controls whether the PB0 pin sources 100  $\mu$ A or sinks at least 1 mA. When the ISEN bit is clear, the PB0 pin behaves the same as a standard I/O port pin.

**Table 4. Current Source/Discharge Control**

ACR control bits		Current source	Discharge device	Action
ISEN	CHG			
0	X	OFF	OFF	Current source and discharge device both disabled
1	0	OFF	ON	Discharge current into PB0
1	1	ON	OFF	Source current from PB0

As shown in the block diagram in [Figure 1](#), the current source, discharge device, and the positive input of comparator 2 all share PB0. Therefore, this current source can be used to supply current to external devices which the comparator can then test against some other voltage connected to its negative input through the channel select bus.

If PB0 is left unconnected, it appears like a 20-pF capacitor which requires about 800 nanoseconds to charge from  $V_{SS}$  to 4 volts.



## Current Source

The source current can vary as much as 10% due to the combination of temperature ranging from  $-40\text{ }^{\circ}\text{C}$  to  $+85\text{ }^{\circ}\text{C}$ , supply voltages within  $\pm 15\%$ , and between different units. If the user wants a closer tolerance in the source current, then one or more of the various effects needs to be constrained to a tighter range.

The typical effects are:

1. A 10% change in the supply voltage typically creates about a 1.5% change in the source current.
2. A temperature drop from  $25\text{ }^{\circ}\text{C}$  to  $-40\text{ }^{\circ}\text{C}$  typically creates about a 1% change for each  $10\text{ }^{\circ}\text{C}$  change in temperature.
3. A temperature increase from  $25\text{ }^{\circ}\text{C}$  to  $85\text{ }^{\circ}\text{C}$  typically creates about a 0.01% change for each  $10\text{ }^{\circ}\text{C}$  change in temperature.
4. As the voltage on PB0 changes, non-linearity creates up to about 1% variation in the nominal  $100\text{ }\mu\text{A}$  source current.
5. Variations between different devices can typically have about 1% variation in the source current.

This source supplies a constant current out of PB0 at voltages from  $V_{SS}$  to within 0.5 volt of  $V_{DD}$ . Over the common mode range of  $V_{SS}$  to  $V_{DD} - 1.5$  volts, this current is constant within 1% of the nominal  $100\text{-}\mu\text{A}$  value. The current source continues to work above the common mode range, but starts to degrade below its nominal value as shown in [Figure 10](#).

The internal current source circuitry contains a similar current mirror with bias currents totaling about  $120\text{ }\mu\text{A}$ . To prevent the current source from lagging when the PB0 pin is being switched between sourcing or discharging current, it is either directed out of the PB0 pin or redirected to  $V_{SS}$ . Otherwise, the lag in the current source would affect the timing of single slope A/D conversions. Therefore, the current source draws about  $220\text{ }\mu\text{A}$  whenever the ISEN bit is set, even if the PB0 pin is not sourcing current and the discharge device is active. As the voltage on PB0 changes, non-linearity creates up to about 1% variation in the nominal  $100\text{-}\mu\text{A}$  source current.

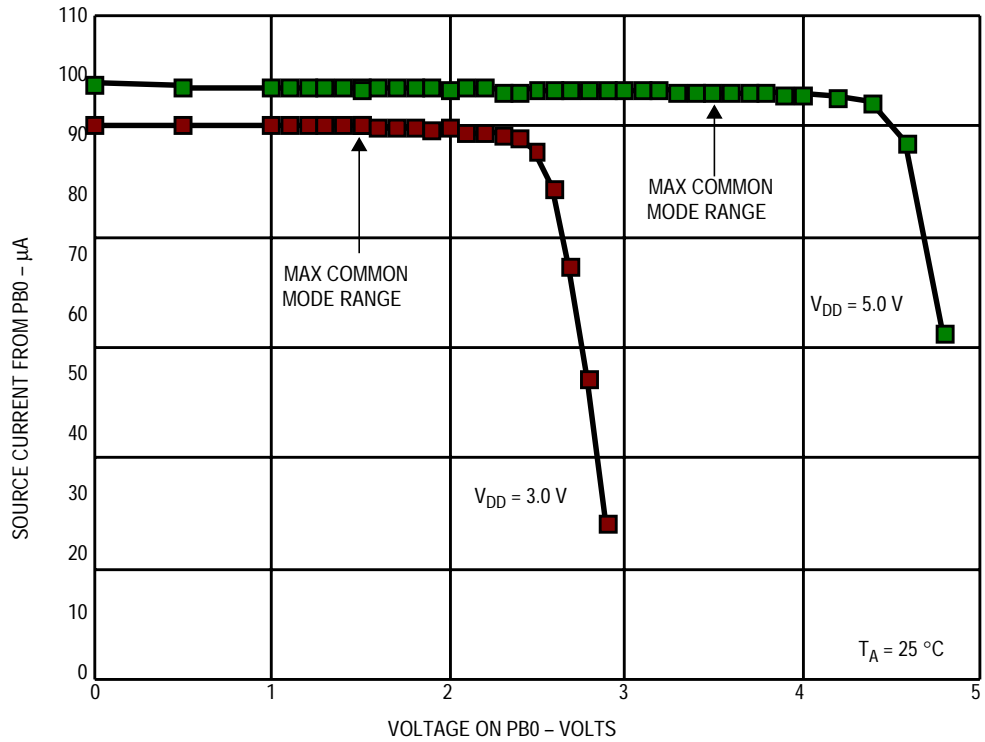
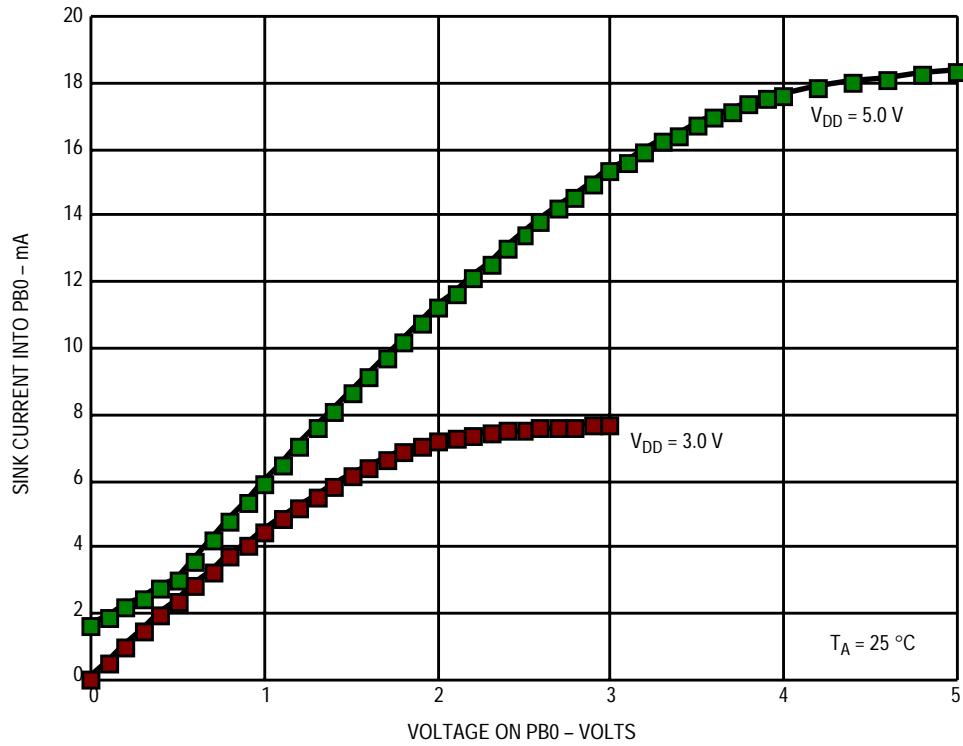


Figure 10. Typical Current Source Behavior (PB0)

Discharge Device

Also attached to PB0 is a MOS transistor designed to sink at least 1 mA of current from a source of 0.4 volts into V<sub>SS</sub>. The design goal was to provide a discharge current at least 10 times the source current. However, this device does not behave like a pure resistor as its current sinking capability varies with both the supply voltage and the voltage present on the PB0 pin.

The typical discharge current versus PB0 voltage and supply voltage is shown in Figure 11. For V<sub>DD</sub> of 3 volts, the equivalent discharge resistance is about 250 Ω, and for V<sub>DD</sub> of 5 volts, the equivalent discharge resistance is about 190 Ω.



**Figure 11. Typical Discharge Current Behavior (PB0)**

Current spikes can occur when discharging an external capacitor through PB0, which can be disruptive to the internal analog V<sub>SS</sub> with respect to the external V<sub>SS</sub> pin.

**NOTE:** *Discharging large capacitors or low impedance loads into the PB0 pin can cause high current spikes which may disrupt the internal analog  $V_{SS}$  with respect to the  $V_{SS}$  pin and cause unwanted analog noise. Repeated high currents can also damage the discharge device and/or the device's interconnect metal. Using capacitors that are less than  $2\ \mu\text{F}$  and discharge current peaks that are less than  $25\ \text{mA}$  is recommended.*

The current source and discharge device is an important part of building a single slope A/D convertor. In conjunction with A/D conversions, the ATD1 and ATD2 mode bits in the ACR can provide some automatic switching between the current source and sink modes of operation.

## Analog Signal Multiplexers

---

The analog signal multiplexer is shown in [Figure 12](#). It is controlled by the bits in the AMUX register at location \$0003. By design, this register has minimal interlocking between some of these control bits. The reason for this is to allow combinations of channels to perform such tasks as current summing and load switching.

The analog signal multiplexer is divided into two sections:

- The channel select bus
- The comparator input bus

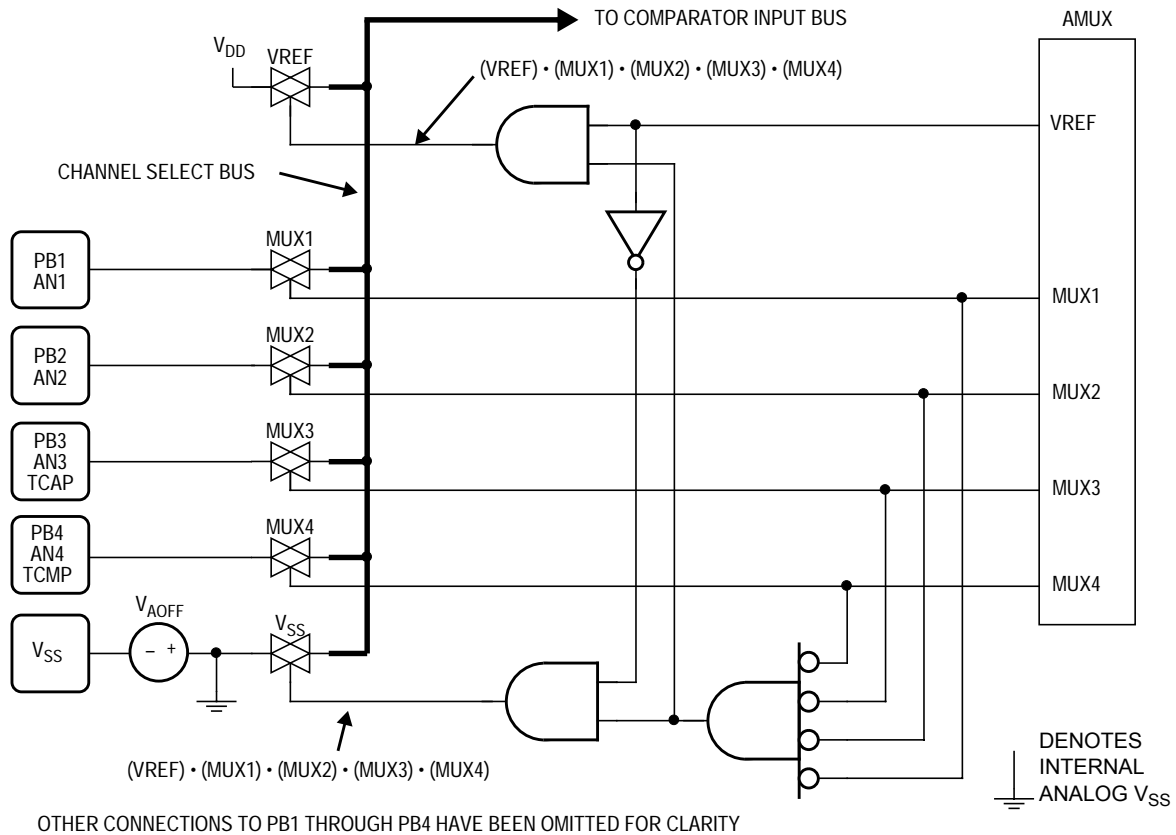


Figure 12. Analog Channel Bus

Channel Select Bus

The VREF, MUX1, MUX2, MUX3, and MUX4 bits in the AMUX register control which signals or pins are connected to the channel select bus. The MUX1:MUX4 bits can select any or all PB1:PB4 pins with no interlocks and connect them to the channel select bus. When none of the MUX1:MUX4 bits are set, the channel select bus is connected to either the internal  $V_{DD}$  or  $V_{SS}$  depending on the state of the VREF bit. If any MUX1:MUX4 bit is set again, then it opens the  $V_{DD}$  or  $V_{SS}$  connection regardless of the state of the VREF bit.

**NOTE:** *The simple channel multiplexers do not include any automatic break-before-make or make-before-break control of multiple simultaneous changes. Therefore, the user must control the multiplexers in the proper sequence to avoid shorting two channels together during the transition. Unless the user has determined that switching transients are not a*

*problem, it is recommended to avoid changing more than one MUX bit on a given write cycle to the AMUX register.*

The channel multiplexers are always tied to the port pins regardless of the programming of pin pulldowns or the digital interface. Therefore, the channel multiplexers can monitor inputs to the digital interface. Similarly, the digital interface can feed outputs to the channel multiplexers, if desired.

### Comparator Input Bus

One end of the comparator input bus connects to the negative input of comparator 2. The other end of the comparator input bus can connect directly to the channel select bus, through a divider to the channel select bus or to the internal temperature sensing diode. A sample capacitor of approximately 10 pF is always connected to the negative input of comparator 2 regardless of what source is selected. The reference end of this sample capacitor is connected either to  $V_{SS}$  or to an offset voltage depending on the state of the VOFF bit in the ASR. A block diagram of the comparator input bus is shown in [Figure 13](#).

The HOLD and DHOLD bits in the AMUX register control the comparator input bus as given in [Table 5](#). The normal mode is to have the HOLD bit set and the comparator input bus connected directly to the channel select bus. However, this limits input sources to the common mode range. When performing A/D conversions, the channel select bus needs to be tied to a reference voltage such as  $V_{DD}$  to make a  $V_{REF}$  conversion. To select  $V_{DD}$ , the divided input is needed by selecting the DHOLD bit instead of HOLD bit. When the divided input is used, the source could be as high as  $2 \times V_{DD} - 3$  volts and still not exceed the common mode range of the comparator. However, the input protection diode on the port B inputs will try to limit the voltage on the pins to  $V_{DD} + 0.3$  volts. Supplying a higher voltage than  $V_{DD} + 0.3$  volts could damage this diode and the device. The illogical case of setting both the HOLD and DHOLD bits does not actually short out the divider, but selects the temperature diode instead.

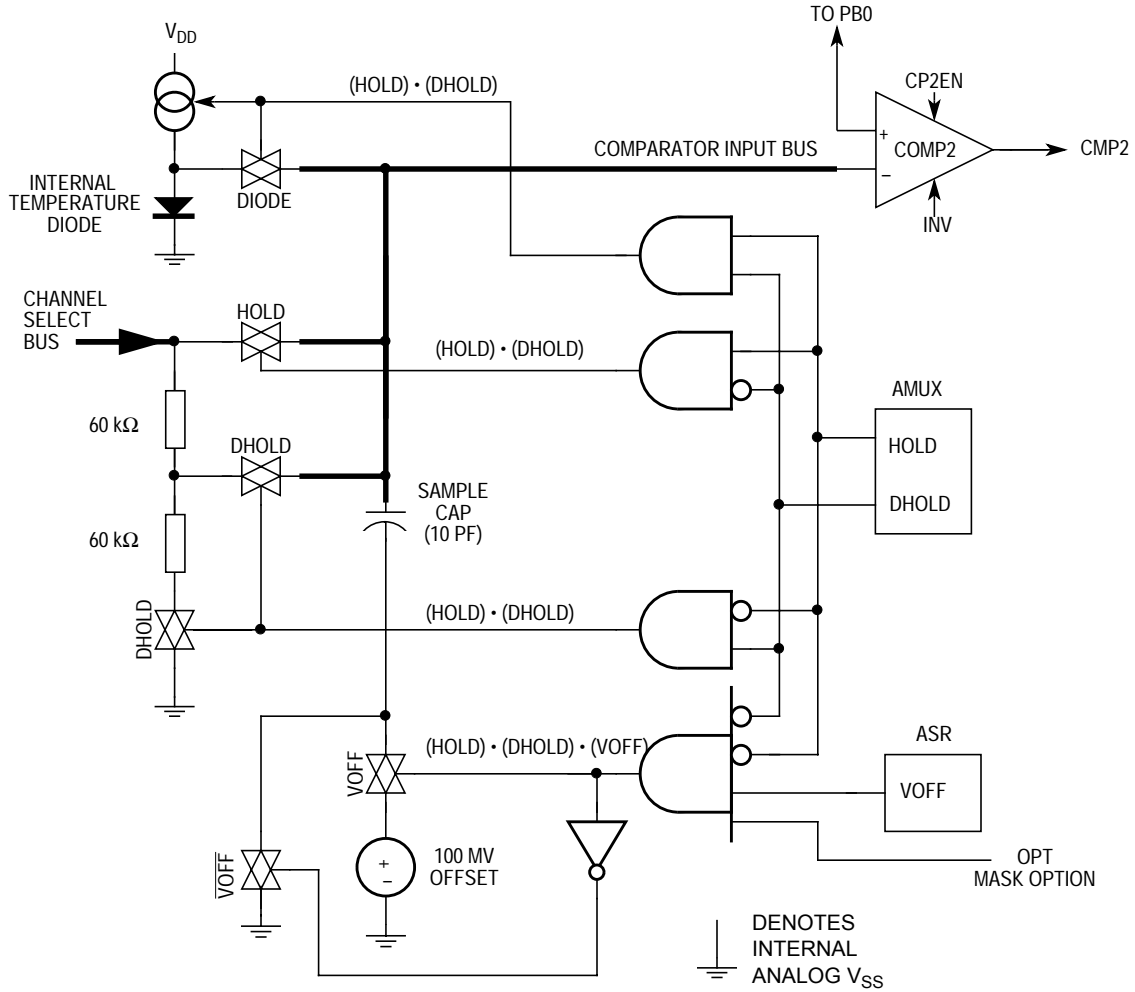


Figure 13. Comparator Input Bus

Table 5. Comparator Input Bus Selections

AMUX register		Input source
HOLD	DHOLD	
0	0	Open circuit; hold voltage on sample capacitor
0	1	Divided input from channel select bus
1	0	Direct input from channel select bus
1	1	Connected to internal temperature diode

## Application Note

Multiplexer  
Impedance

All the internal multiplexers are made identical for reasons of balancing the impedances seen by the input divider. The impedance of the multiplexers is not a simple resistive switch, but a complex transfer characteristic from the MOS transistors used in the multiplexer. Therefore, this impedance increases as the supply voltage decreases, the voltage across the multiplexer increases, or the device temperature increases. The actual impedance of each multiplexer is difficult to precisely determine, but the typical impedance behavior of a pair of multiplexers in series is shown in [Figure 14](#). This impedance can be modeled as a resistor of 800–1500  $\Omega$  at a  $V_{DD}$  of 5 volts and 1200–2900  $\Omega$  at a  $V_{DD}$  of 3 volts.

When the channel select bus is connected directly to the input of comparator 2, the multiplexer impedance is not an issue since the comparator's input impedance is several orders of magnitude greater. The only effect would be a minor increase in the charge time of the sample capacitor. When the divided input is used, a compensation multiplexer is added to the bottom of the divider to track the impedance of the channel input multiplexer and keep the overall divider ratio at 0.5. However, this dummy multiplexer at the bottom of the divider cannot compensate for multiple input channels being on in parallel.

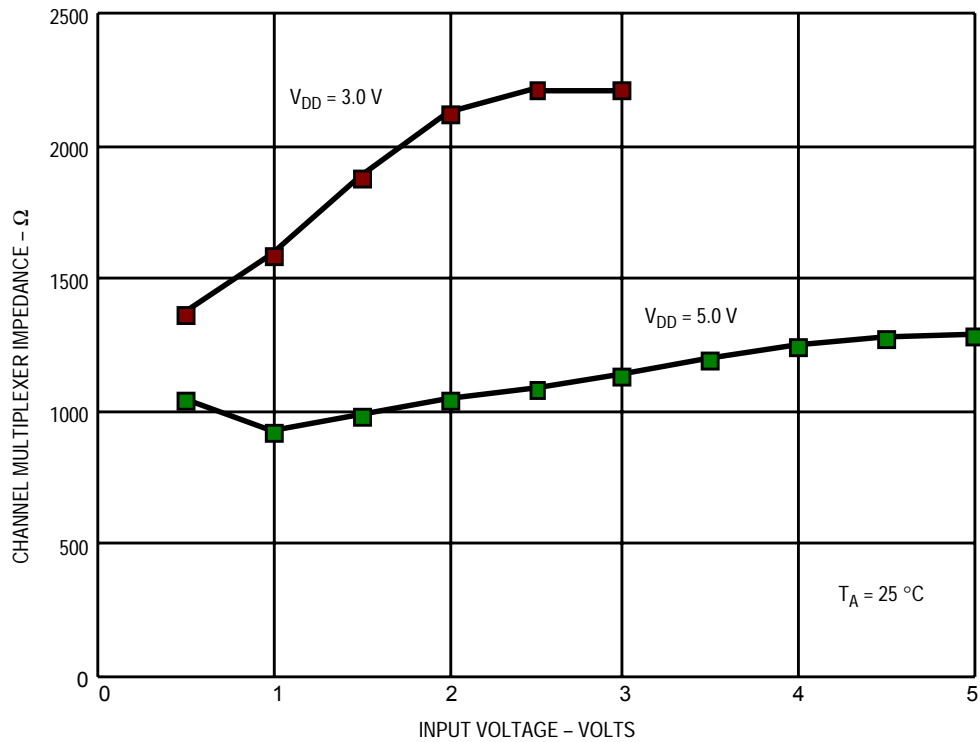
**NOTE:** *The divided input is only compensated for one channel multiplexer at a time. If multiple channel multiplexers are on simultaneously, then the divider ratio will increase slightly.*

Therefore, the exact impedance of the multiplexers is only a consideration when the user is operating with more than one multiplexer on at a time or is passing current into one channel multiplexer and out of another. A first order estimate can be made by taking the data given in [Figure 14](#) and dividing by two for two channels in parallel, dividing by three for three channels in parallel, and so on.

## Parallel Channels

The ability to parallel input channels can be used to sum currents as shown in [Figure 15](#). In this example, two current sources,  $I_1$  and  $I_2$ , are fed into the channel select bus by PB1 and PB2. Their summed current then flows out of PB3 to a voltage conversion resistor, R.





**Figure 14. Typical Channel Multiplexer Impedance**

If PB1 and PB2 are being fed from current sources and sufficient voltage range is available, then the impedance of the MUX1 and MUX2 channel multiplexers does not create a voltage drop relative to the internal channel bus. The voltage on the internal channel bus with PB1, PB2, and PB3 all on will be:

$$V_{\text{CHANNEL}} = (I_1 + I_2) \times (R + Z_{\text{MUX}})$$

If the PB4 pin is connected to V<sub>SS</sub>, then MUX3 can be turned off and MUX4 turned on in its place. The voltage on the channel bus will now be:

$$V_{\text{CHANNEL}} = (I_1 + I_2) \times (Z_{\text{MUX4}})$$

Assuming that Z<sub>MUX3</sub> is approximately equal to Z<sub>MUX4</sub>, the difference between these two voltages, therefore, is the voltage on the external conversion resistor, and the effects of the channel multiplexer’s impedances are cancelled. With this setup, the currents through port pins should be limited to less than 1 mA.

Application Note

For the widest operating range, the currents and conversion resistor should be selected such that the total voltage drop through two multiplexers and the conversion resistor is less than the common mode range for the comparators.

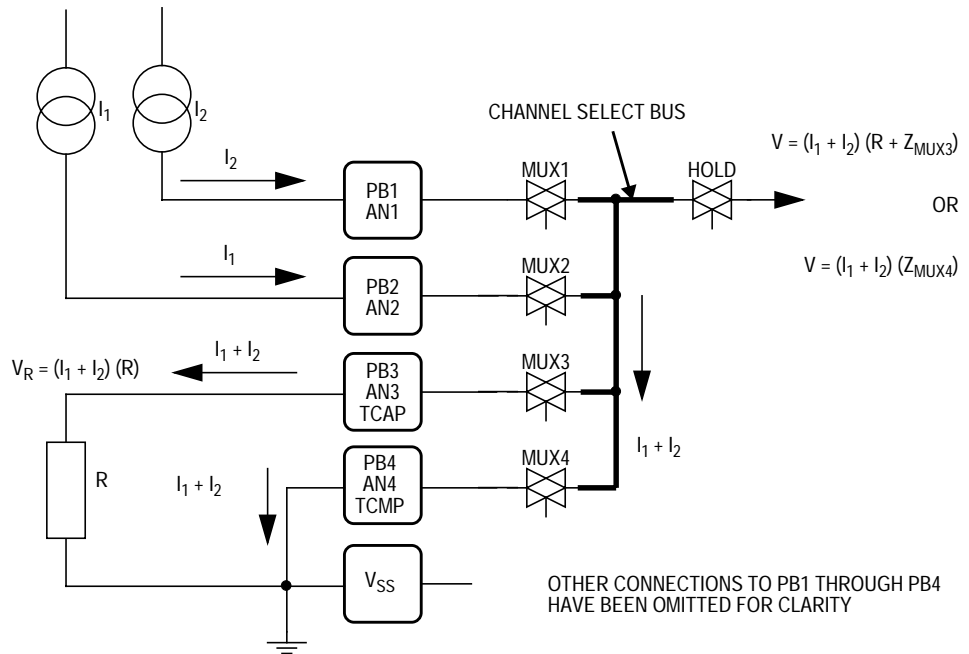


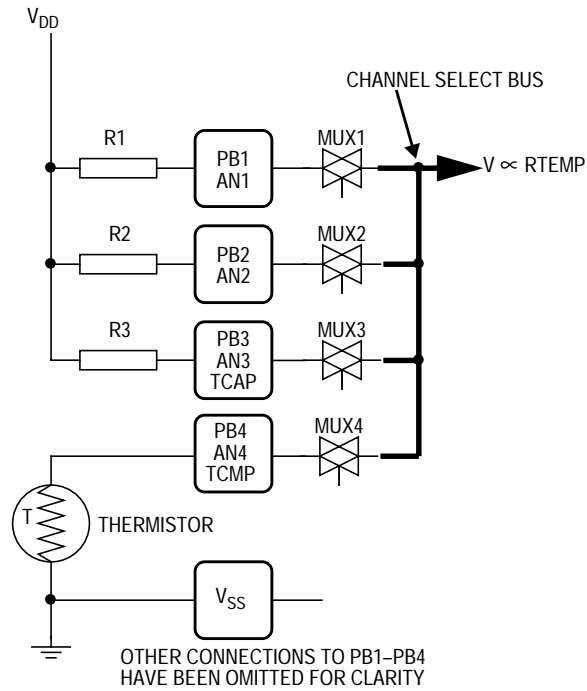
Figure 15. Current Summing Inputs

The multiplexer channels also can be connected to supply a source voltage to an external device as shown in Figure 16. As shown,  $V_{DD}$  can be applied through three different resistors to PB1, PB2, or PB3; through multiplexers MUX1, MUX2, and MUX3; and then out through MUX4 and PB4 to an external thermistor. Each resistor can be selected for a given temperature range of the thermistor. The resulting voltage can be measured on the channel bus. Even though there is a multiplexer on either side of the channel bus, there will be some effects from the temperature variation in the multiplexer impedances. This effect can be minimized at temperatures where the source resistor creates the same voltage drop across both the thermistor and the source resistor.

Freescale Semiconductor, Inc.

**Input Limitations**

Avoid driving input pins below  $V_{SS}$ , even if the channel multiplexers are OFF, as the input protection diodes will start turning on at about  $-0.3$  volts and inject high currents through the  $V_{SS}$  pin. Similarly, avoid driving the input pins more than 0.3 volts above the  $V_{DD}$  supply level.



**Figure 16. External Source Control**

## Support Circuitry

A number of specific support circuits provide additional capabilities or options as described in the next subsections.

### Input Divider

The primary purpose of the input divider is to extend the common mode range to allow measurement of pin voltages up to the  $V_{DD}$  supply level. This divider can be switched in between the channel select bus and the comparator input bus by setting the DHOLD bit and clearing the HOLD bit in the AMUX register. The input divider can be used only on the channel select bus signals and cannot be used in conjunction with the temperature diode. The divider is created with two identical resistances, each of a nominal  $60\text{ k}\Omega$ . This provides a divider ratio of 0.5. The matching of these resistors is within  $\pm 1\%$ , which results in the divider ratio varying from 0.49 to 0.51 between different devices.

The input impedance of the comparator is more than  $1\text{ M}\Omega$ , so the divider becomes a significant input impedance load when switched into the signal path. At DC, this input impedance looks like  $120\text{ k}\Omega$ , but at frequencies above  $10\text{ kHz}$  the  $10\text{ pF}$  sample capacitor starts to bypass the lower leg of the divider and both the input impedance degrades and the divider ratio starts to drop. Neglecting the parasitic capacitances in the multiplex circuitry, the input impedance can be roughly modeled as shown in **Figure 17** with typical input impedance as shown previously in **Figure 9**.

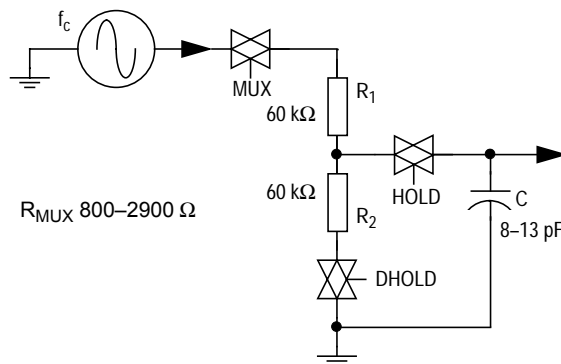


Figure 17. Divider Equivalent Circuit

To cancel the impedance of the channel multiplexers, an identical dummy multiplexer is switched in at the bottom whenever the DHOLD bit is set. If more than one channel multiplexer is turned on at the same time, there will be a shift in the divider ratio due to the parallel combination of the input multiplexers not being matched by the compensation multiplexer at the bottom of the divider. A second-order temperature-dependent term also will be present if the multiplexer impedance at the top is not equal to the dummy multiplexer at the bottom. Typical input impedance of both the direct and divided inputs is given previously in [Figure 9](#).

**NOTE:** *The divided input is only compensated for one channel multiplexer at a time. If multiple channel multiplexers are on simultaneously, then the divider ratio will increase slightly.*

#### Sample and Hold

The sample capacitor is always connected from the negative input of comparator 2 to  $V_{SS}$ . It is typically 10 pF, but can vary from 8 to 13 pF between different devices. The capacitor can be charged within 99.9% of its final value within seven RC time constants.

When connected directly to the channel select bus, the worst case series impedance of the channel and the HOLD multiplexers is about 8 k $\Omega$  for an RC time of 104 nanoseconds. By comparison, it takes the software four CPU bus cycles to set and then turn off the HOLD bit. At the maximum oscillator frequency of 4.2 MHz, this is about a 2-microsecond delay or about 18 RC time constants. Therefore, if connected directly to the channel bus, there should always be sufficient time to charge the sample capacitor between any two instructions.

However, when connected through the divider to the channel bus, the series impedance of the channel and the DHOLD multiplexers is added to the 60-k $\Omega$  equivalent resistance of the divider. This increases the RC time constant to a worst case of 832 nanoseconds. With the same 2-microsecond software delay, this would give a little more than two RC time constants, which would allow only the sample capacitor to charge to about 85% of its final value. Therefore, if connected through the divider to the channel select bus, there should always be sufficient time to charge the sample capacitor (seven RC time constants or about six

microseconds). If four NOP instructions are inserted between the instructions that set and clear the DHOLD bit, then this will add an additional eight CPU cycles which, when combined with the original four CPU cycles, will give an equivalent delay of about six microseconds. If insufficient time is given to allow the sample capacitor to fully charge (or discharge) to a new voltage, then such measurements will be in error.

**NOTE:** *It is important that the INV and VOFF bits are not changed during the time that a voltage is being held on the sample capacitor. The exchange of inputs and switching currents can easily corrupt, if not completely discharge, the sampled voltage.*

The sample capacitor typically can hold its charge for several minutes at room temperature, but due to inherent leakage currents, this degrades quickly as the temperature is raised. Leakage currents in silicon double every 10 °C such that a minute decay time can degrade to less than a second at 85 °C. In practice, the decay rate of 0.2 V/second should be used for the overall temperature range of –40 °C to + 85 °C.

**NOTE:** *The window on erasable EPROM devices must be covered with an opaque material such as black electrical tape. Common white paper labels are not opaque enough to prevent light from affecting bias currents, trip points, and leakage in the analog subsystem.*

One observed source of sample corruption comes from  $V_{SS}$  ground bounce noise, which tends to disturb the multiplexers attached to the sample capacitor. If they turn on, even for a few nanoseconds, the voltage to which they are connected, either higher or lower, will be connected to the sample capacitor and will alter its charged voltage. Input signals which remain connected to their source are not affected by these noise spikes signals due to their short durations. In many cases, the user is in control of the significant sources of ground bounce noise. These occur when the MCU's I/O pins suddenly turn on and sink high currents through the  $V_{SS}$  pin. It is recommended that switching of any loads does not occur during the time the sample capacitor must hold its charge. Other sources can be poor layouts.

Another source of sample corruption comes from substrate current injection which can occur when the input protection diodes begin conduction in order to limit inputs which have gone above  $V_{DD}$  or below

$V_{SS}$ . The user also has some control over substrate current injection by limiting voltage and/or current excursions on the pins.

Refer to the application note *In-Circuit and Emulation Consideration for MC68HC05JJ/JP Series Microcontrollers*, Freescale document number AN1741/D, for more information on good PWB layout and limiting these noise source.

One particularly troublesome high current case occurs when using the discharge device on PB0 to discharge an external capacitor at the end of a single slope A/D conversion. While normally there is no need to make multiple A/D conversion measurements from the same sample, a diagnostic may try to do this to measure the leakage performance of the sample capacitor. In this case, the repeated measurements, with discharges into PB0 at the end of each one, will yield worse results than actually occur with a single A/D conversion.

**NOTE:** *Experience has shown that the least sensitivity to such noise sources is created when the channel select bus is connected to the internal  $V_{DD}$  after the HOLD or DHOLD multiplexer has been turned off. This also means that the input channel multiplexers are off. Leaving the channel bus connected to  $V_{SS}$  or a low voltage input creates the most sensitivity to  $V_{SS}$  ground bounce noise.*

**NOTE:** *When trying to maintain the sample voltage, avoid discharging capacitors or other high current loads through the MCU's  $V_{SS}$  pin or power supply return wiring.*

**NOTE:** *In cases where severe system noise affects input signal measurements, it is recommended that the sample and hold method not be used. Instead, always leave the signal connected to the comparator through the HOLD or DHOLD and channel multiplexers.*

Another source of sample errors can occur if the proper sequencing of the multiplexers is not followed. Since there is no automatic control of the multiplexers, the user should first turn off the HOLD or DHOLD multiplexer before changing any channel multiplexers. If all the multiplexers are changed on the same write cycle to the AMUX register, there can occur some charge redistribution due to the tiny variations in delays between all the multiplexers. The only exception to this is when

switching to the temperature diode, where the HOLD and DHOLD must be changed at once and the sample and hold function is not available.

**NOTE:** *When using the sample and hold capacitor, always turn off the HOLD and DHOLD bits on a separate write to the AMUX register before making any changes to the other channel multiplexers.*

When comparator 2 is not being used to measure a voltage, it should have its sample capacitor and input tied through the HOLD multiplexer to the channel select bus which has been connected to  $V_{SS}$ . Otherwise, the input on comparator 2 may drift with stray leakage currents and cause unwanted output changes or flags.

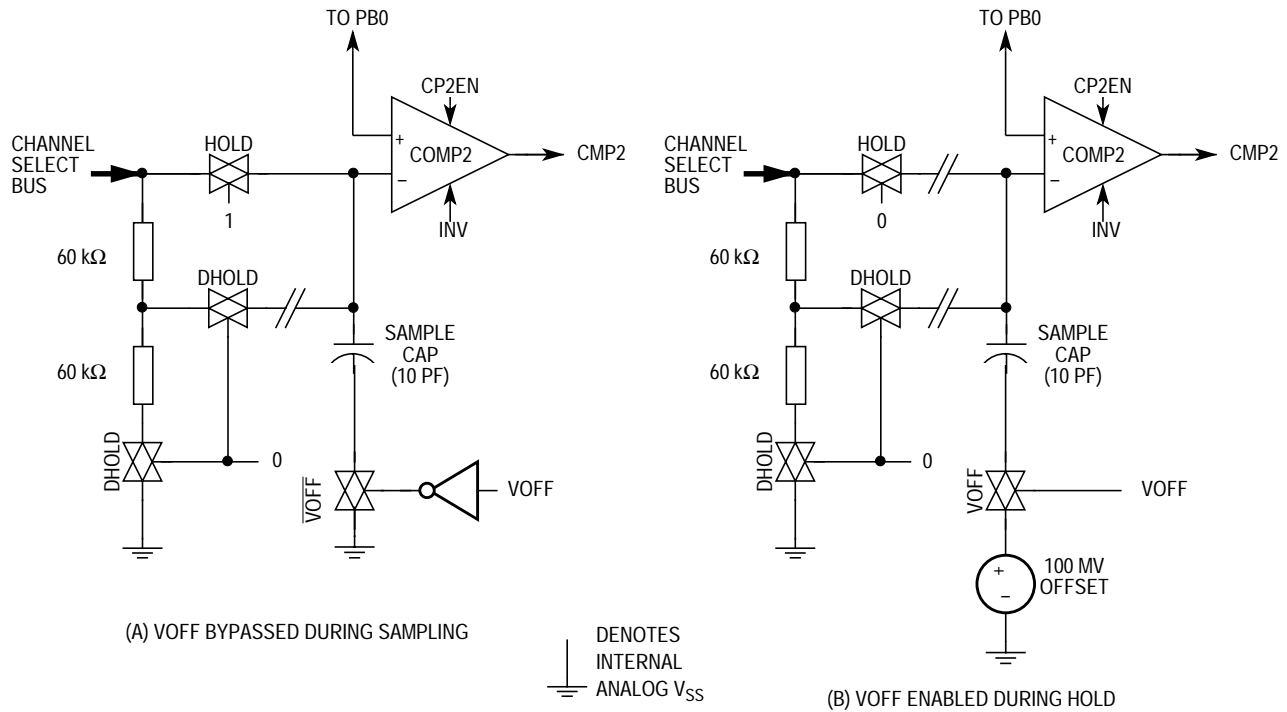
#### Sample Offset Voltage

As the input voltages get near the  $V_{SS}$  level, the magnitude and phase of the comparator's own offset voltage can create a situation where an external pin must go below  $V_{SS}$  in order to flip the comparator output. To handle this case, the VOFF bit can add a small positive offset voltage (nominally 100 mV) to the voltage stored on the sample capacitor as shown in [Figure 18](#).

During the sampling of the input pin, the lower end of the sampling capacitor is connected directly to the internal  $V_{SS}$  line. The sampling ends when both the HOLD and DHOLD bits are off. In this case, the sampling capacitor has the input voltage with respect to  $V_{SS}$ , and the lower end of the sampling capacitor is raised about 100 mV above  $V_{SS}$ . Now any input offset in the comparator will be added to or subtracted from the nominal 100 mV.

The VOFF bit is not active unless the OPT bit is selected as a mask option. About 20  $\mu\text{A}$  of  $I_{DD}$  current is drawn when the VOFF bit is set, and the general analog bias source must be powered up. This can be done by setting either the ISEN, CP1E, or the CP2E bits in the ACR.





**Figure 18. Sample Voltage Offset Source**

**Temperature Diode**

If some measurement of temperature could be made, the MCU software could correct for some of its temperature effects on the current source, low-power oscillator, or MUX impedances. Such a temperature measuring device also could be used to monitor the device's temperature when one or more I/O pins is used to supply power to external loads.

A simple PN diode is provided for this purpose. It is connected to the internal  $V_{SS}$  and can be used to make such measurements of the temperature of the device. This diode is next to the current source which is located near the PB1 pin. The voltage on this diode is typically 0.7 volts. To create this voltage, about 20  $\mu A$  of current are supplied when both the HOLD and DHOLD bits are set. For this 20  $\mu A$  to be supplied, the general analog bias source must be powered up. This can be done by setting either the ISEN, CP1E, or CP2E bits in the ACR. Once powered, the diode's voltage will change approximately 2 mV/ $^{\circ}C$ .

The diode voltage drop and change with temperature are not accurate enough to measure any closer than 10 °C, unless the device is calibrated with a table of values versus temperature that is stored in the personality EPROM or user EPROM found on the MC68HC705JJ7 or MC68HC705JP7.

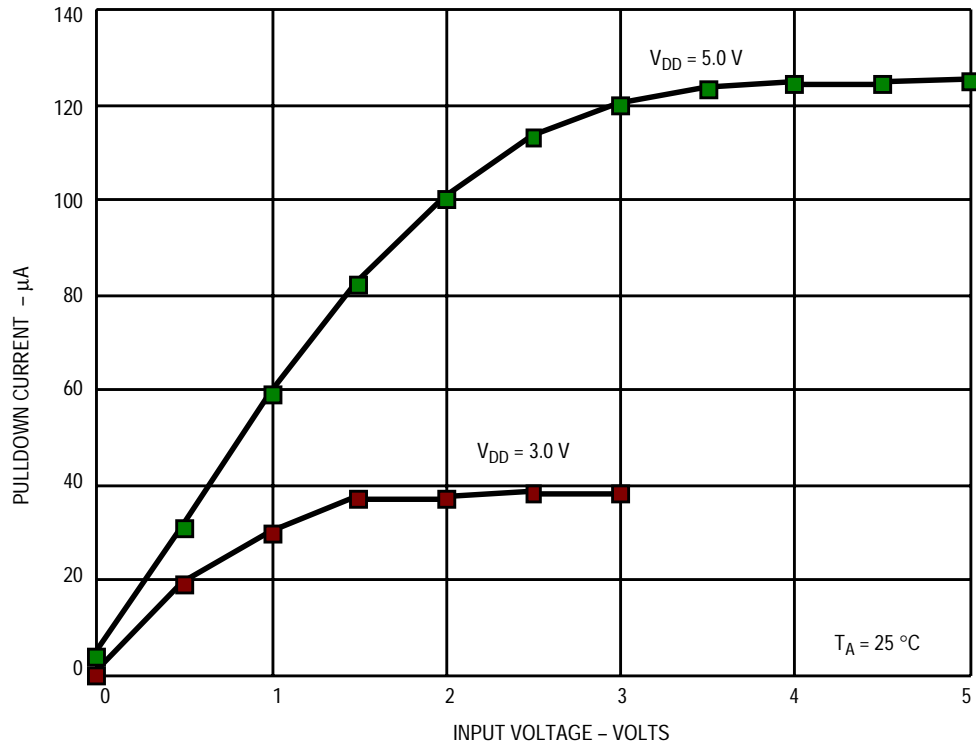
## Input Pulldowns

External pulldown resistors are normally needed for inputs to switches or other high impedance sources. To reduce cost, it is advantageous to supply these pulldown currents internally in the MCU. The JJ/JP series has pulldowns, which can be controlled by software, on all pins on ports A, B, and C.

As the default condition following a reset or power-on condition, these pulldowns come up activated so that all I/O port inputs will be immediately defined. If the user does not want these pulldowns to be present during power on, they can be disabled by the SWPDI bit in the mask option register (MOR) or by a mask option on ROM-based devices. These pulldowns also can be individually disabled by software which writes to the PDRA and PDRB registers at \$0010 and \$0011, respectively.

These pulldowns are intended for digital functions to eliminate external resistors and are actually weak N-channel MOS devices connected to  $V_{SS}$ . Because they are MOS devices, they have non-linear pulldown behavior as shown in [Figure 19](#).

**NOTE:** *Be sure to turn off any pulldowns on the PB0:PB7 pins when these are used for analog signals. Leaving these pulldowns on, especially on PB0, can lead to unwanted analog parasitics and unexpected measurement results. When the pulldown for PB0 is on when charging an external capacitor, the ramp will look like an RC charge curve rather than a linear ramp.*



**Figure 19. Typical Input Pulldown Current**

**16-Bit Timer  
Input Capture**

The 16-bit timer is the same one found on some other members of the MC68HC05 Family of MCUs. In the JJ/JP series, it has an additional purpose in providing the A/D timing in mode 2 or mode 3. It may be difficult to use the input capture and output compare timing functions for external events while still using them for the A/D conversions, but careful planning of the A/D sampling and conversion times will provide more capability for activities to be done in parallel.

When using the 16-bit timer for an A/D mode 2 or mode 3 conversions, the ICIE and IEDG bits in the TCR must be set in addition to the ICEN bit in the ACR. This is because the input capture function must be enabled and set up to trigger on the rising output of comparator 2. If one or more of these bits is left cleared, A/D conversions will not take place.

The ICF, OCF, and TOF timer flags interact to start and stop the current source driving PB0. Under this control, the ICF has priority over both the OCF and TOF. Further, the latched static flag, CPF2, of comparator 2 is

used to drive the input capture. Therefore, take care to follow these rules:

1. Always clear the TOF and OCF before clearing the ICF. Otherwise, clearing the ICF immediately will enable any existing TOF or OCF to restart the charging cycle without regard for their proper timing points.
2. Avoid clearing the CPF2 comparator static flag too quickly after clearing the ICF as this enables the timer to capture a new time should the comparator output then rise again later.
3. Avoid changing the state of the INV bit after clearing the ICF as this will cause a switching pulse to set the CPF2 bit and trigger an unwanted ICF.
4. Changes to the ICEN, ICIE, or IEDG bits can cause the ICF flag bit to become set. Always clear the ICF flag following any change in the state of the ICEN, ICIE, and IEDG bits.
5. Account for the first TOF, OCF, and ICF flags which can get set without any enabling control. They can get set *before* the software is ready to use them.

The 16-bit timer continues to run while in wait mode, and this allows the mode 2 or mode 3 A/D conversions to take place during wait mode. When the conversion is complete, a timer interrupt will occur and the MCU will exit wait mode.

### Analog Interrupt

A separate analog interrupt is provided with a vector location at \$1FF2 and \$1FF3. This interrupt is triggered by either the CPF1 or CPF2 flags being set. These bits latch so that the triggering comparator can be determined by polling the CPF1 or CPF2 bits in the analog interrupt service routine. Since these latch bits are the triggering mechanism, they should both be cleared before exiting the analog interrupt service routine.

## Analog Power-Up Considerations

---

All of the analog subsystem components can be powered down as an  $I_{DD}$  current saving technique. The controls for each comparator and the current source are contained in the ACR at location \$001D. Internally, there is an analog bias supply which provides bias currents for all of the analog features. When either comparator or the current source is powered up, the bias supply is turned on automatically.

The maximum time delay for an analog component to power up and stabilize is five microseconds. This delay is for individual components. If several of the analog components are all initially powered up on the first write cycle to the ACR, then the delay may be as long as 10 microseconds as both the bias supply and the comparators must come up together.

**NOTE:** *Before sampling the comparator outputs or clearing the CPF1 or CPF2 flags, be sure to allow sufficient delay after the comparators are powered up. Otherwise, the outputs may be undefined or the flags may not be cleared.*

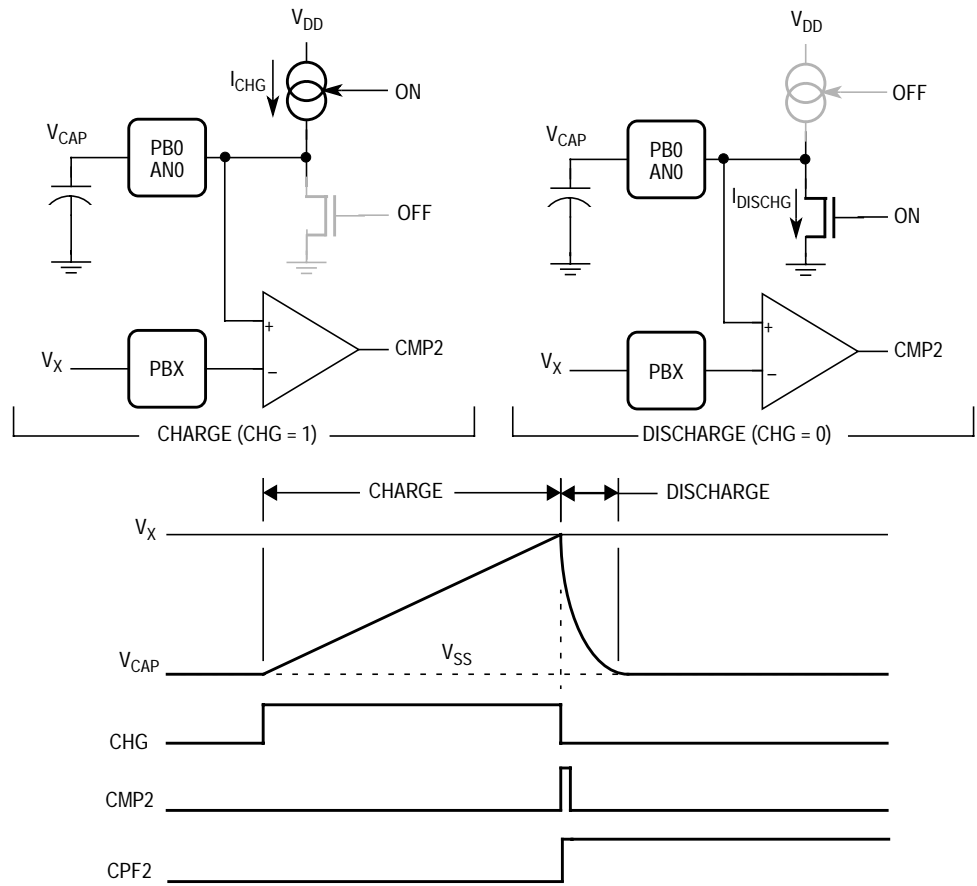
## Analog-to-Digital Conversion

---

Analog-to-digital (A/D) conversion can be accomplished using the JJ/JP series by allowing the current source to charge up (ramp) an integration capacitor connected to PB0 and using comparator 2 to detect the time when the ramp voltage crosses the unknown voltage being measured as shown in [Figure 20](#).

Application Note

Freescale Semiconductor, Inc.



**Figure 20. Single Slope A/D Converter**

The advantages of this design are:

- Simple construction
- Minimal on-chip silicon compared to most 8-bit successive approximation register (SAR) A/D designs
- More flexible than an SAR A/D, which must always perform as an A/D and require its own D/A conversion circuitry
- Some “filtering” of input signal provided by the integration
- Inherently monotonic
- Up to 16 bits are possible with long conversion times

On the other hand, single slope A/D convertors have some disadvantages:

- Slow conversion times, in the milliseconds
- More software required to do the actual conversion
- External ramp capacitor needed

Overall, the single slope A/D is well suited to many applications with slow-moving signals (<1000 Hz) which are common with many analog sensors and for user input sources like potentiometers. The additional software is offset by the smaller size of the single slope A/D circuitry. The single-slope A/D support circuitry on the MC68HC05JJ/JP series of MCUs uses about 20% less silicon area than the 8-bit SAR A/D commonly found on the MC68HC05P series; and this 20% reduction is equivalent in silicon area to about 400 bytes of ROM. Therefore, even with extensive software, the single slope A/D implementation will use less silicon area than a typical SAR A/D design.

**Performance Calculations**

Fundamental to the single slope A/D conversion technique is the equation relating time to voltage dependent on the external ramping capacitor and its charging current:

$$t_{\text{CHG}} = \frac{C_{\text{EXT}} \times V_X}{I_{\text{CHG}}}$$

where:

$t_{\text{CHG}}$  = Charge time (seconds)

$V_X$  = Voltage to reach (volts)

$C_{\text{EXT}}$  = External ramp capacitor ( $\mu\text{F}$ )

$I_{\text{CHG}}$  = Charge current ( $\mu\text{A}$ )

From this basic equation, the time can be translated into a number of counts at a given frequency:

$$t_{FS} = \frac{P \times N_{FS}}{f_{OSC}} = \frac{C_{EXT} \times V_{FS}}{I_{CHG}}$$

where:

$t_{FS}$  = Full scale charge time (seconds)

$N_{FS}$  = Full scale counts (counts)

$P$  = Prescaler (counts out/counts in)

$f_{OSC}$  = Oscillator frequency (Hz)

$V_{FS}$  = Full scale voltage (volts)

From this relationship, the value of each component in the single slope A/D can be defined at the full-scale voltage input as:

$$V_{FS} = \frac{P \times N_{FS} \times I_{CHG}}{C_{EXT} \times f_{OSC}}$$

$$C_{EXT} = \frac{P \times N_{FS} \times I_{CHG}}{f_{OSC} \times V_{FS}}$$

$$N_{FS} = \frac{C_{EXT} \times V_{FS} \times f_{OSC}}{P \times I_{CHG}}$$

The general process is to begin with the desired full-scale voltage, resolution, and speed of conversion. This information will give  $V_{FS}$ ,  $N_{FS}$ , and  $t_{FS}$ . Since the charge current,  $I_{CHG}$ , is fixed at 100  $\mu$ A, the only remaining variables are the size of the external capacitor and the effective clock rate which is the product of the prescaler,  $P$ , and the oscillator frequency,  $f_{OSC}$ . Usually, the clock frequency is fixed by other system factors, but the prescaler can be selected as eight for mode 2 or mode 3 and greater than eight for software routines using mode 0 or 1.



The last variable is the external capacitor size. But, capacitors come in finite sizes, so there is a trade-off in operating frequency, prescaler, and capacitor size that yields the desired count resolution for a given full-scale voltage range. There is no quick rule to select the best combination, as there are obviously several considerations.

The typical range of single slope A/D performance using the MC68HC05JJ/JP series MCUs is given in **Table 6** as a guide to selecting a starting point.

**Table 6. A/D Conversion Parameters**

Name	Function	Min	Typ	Max	Units
$V_X$	Unknown voltage on channel selection bus	$V_{SS}$	—	$V_{DD} - 1.5$	V
$V_{CAP}$	Charging voltage on external capacitor	$V_{SS}$	—	$V_{DD} - 1.5$	V
$V_{MAX}$	Maximum charging voltage on external capacitor	—	—	$V_{DD} - 1.5$	v
$I_{CHG}$	Charging current on external ramping capacitor $V_{DD} = 3$ Vdc $V_{DD} = 5$ Vdc	75 85	90 100	104 113	$\mu$ A
$I_{DIS}$	Discharge current on external ramping capacitor $V_{DD} = 3$ Vdc $V_{DD} = 5$ Vdc	1.0 1.1	— —	— —	mA
$t_{CHG}$	Time to charge external capacitor ( $100$ kHz $< f_{OSC} < 4.0$ MHz) 4-bit result 6-bit result 8-bit result 10-bit result 12-bit result	0.032 0.128 0.512 2.048 8.192	0.128 0.512 2.048 8.196 32.768	2.56 10.24 40.96 120 <sup>(1)</sup> 120 <sup>(1)</sup>	ms
$t_{DIS}$	Time to discharge external capacitor, $C_{EXT}$	—	5	10	ms/ $\mu$ F
$C_{EXT}$	Capacitance of external ramping capacitor	0.0001	0.1	2.0	$\mu$ F
N	Number of counts for $I_{CHG}$ to charge $C_{EXT}$ to $V_X$	1	1024	65536	counts
P	Prescaler into timing function ( $\div P$ ) Using core timer Using 16-bit programmable timer Using software loops	8 8 28	8 8 user de ned	8 8 user de ned	$\div P$
$f_{OSC}$	Clock source frequency (excluding any prescaling) $V_{DD} = 3$ Vdc $V_{DD} = 5$ Vdc	— —	— —	2.1 4.2	MHz

1. Limited for  $C_{EXT}$  to be less than 2.0  $\mu$ F.

All the calculations so far have dealt with variables that are assumed to be fixed, when in reality there are tolerances on  $f_{OSC}$ ,  $C_{EXT}$ , and  $I_{CHG}$ . The worst-case stack up of these tolerances should be considered when calculating  $N_{FS}$  for a given  $V_{FS}$ . Provision must be made to allow the conversion to handle a larger count than would be expected for the ideal case. Otherwise, the count may reach its maximum value too soon, causing a loss of top end capabilities, or the count may not produce enough counts, causing a loss of minimum resolution. A good rule of thumb is to plan on  $N+1$  bits if using an accurate oscillator (crystal or ceramic resonator) and standard external capacitor sizes. If the oscillator is less accurate (internal or external RC) and lower quality external capacitors are used, then plan on  $N+2$  bits. In other words, multiple the range by 2 normally and by 4 for poorer tolerances. When in doubt as to the range of counts possible, figure the minimum and maximum values of  $N_{FS}$  using this calculation:

$$N_{MIN} = \frac{C_{MIN} \times V_{MIN} \times f_{MIN}}{P \times I_{MIN}}$$

$$N_{MAX} = \frac{C_{MAX} \times V_{MAX} \times f_{MAX}}{P \times I_{MIN}}$$

where:

$N_{MIN}, N_{MAX}$	= Min and max full scale count (counts)
$C_{MIN}, C_{MAX}$	= Min and max value of capacitor ( $\mu F$ )
$V_{MIN}, V_{MAX}$	= Min and max full scale voltage (volts)
$f_{MIN}, f_{MAX}$	= Min and max oscillator frequency (Hz)
$I_{MIN}, I_{MAX}$	= Min and max charge current ( $\mu A$ )
$P$	= Prescaler selected (count/count)

With the calculated minimum and maximum counts, the user can determine if they are going to get the desired results.

**NOTE:** *When calculating the minimum and maximum values for  $C_{EXT}$ ,  $V_{FS}$ ,  $I_{CHG}$ , and  $f_{OSC}$ , be sure to include any effects caused by expected changes in supply voltage and temperature.*

## Step-By-Step A/D Conversion

After selecting the right components, the single slope A/D can be operated as a series of simple software controlled steps:

1. Initialize the analog subsystem to the desired configuration.
2. Discharge the internal sample capacitor and channel select bus.
3. Select a signal source.
4. Start charging the ramp capacitor (set CHG bit in ACR).
5. Keep track of time while waiting for comparator 2 output to go high.
6. Capture time when comparator 2 output goes high.
7. Start discharging the ramp capacitor (clear CHG bit in ACR).
8. Calculate the charge time.
9. Compare the charge time to that for a reference signal.
10. When the ramp capacitor is fully discharged, go back to step 3.

Refer to application note *A/D Conversion Software for the MC68HC05JJ/JP Series Microcontrollers*, Freescale document order number AN1739/D, for detailed software examples on how to implement a mode 0, 1, 2, or 3 A/D conversion.

## Pitfalls in Using Single Slope A/D

---

The simplicity of single slope A/D conversion also presents some unwanted faults due to the limits of its input voltage range, signal noise, and timing with respect to the software. The user should consider their application to see if any of the following faults are possible. Four A/D fault conditions can occur in all the modes of operation:

- Infinite conversion time
- Disabled conversions
- Extra conversions
- Early conversions

Two additional A/D fault conditions only occur in mode 2 or mode 3:

- Conversion overlap
- Conversion masking

All six of these faults are described in detail in the following paragraphs.

### Infinite Conversion Time

When the unknown input signal is above the common-mode range of comparator 2, the integration ramp on the PB0 pin can never reach the level to trip comparator 2 and set the CPF2 flag. This is shown in [Figure 21](#).

This fault can be eliminated by making sure that the input signal is clamped within the common-mode range. Another means is to use the divided input which can limit the input to the comparator.

If it does occur, this fault can be detected by making a software check for the maximum conversion time. When this time is reached, the CMP2 flag will still be low, indicating that the ramp never reached the input level. Therefore, the input is at or above the maximum conversion level, the conversion can be terminated, and the conversion result can be forced to its maximum value.

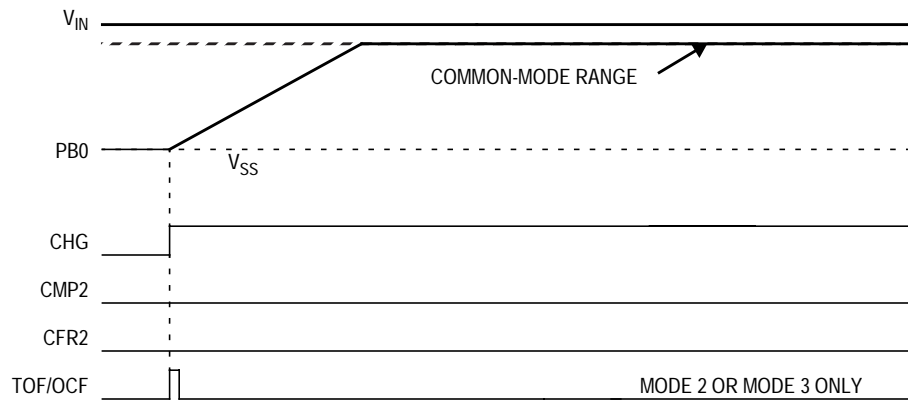


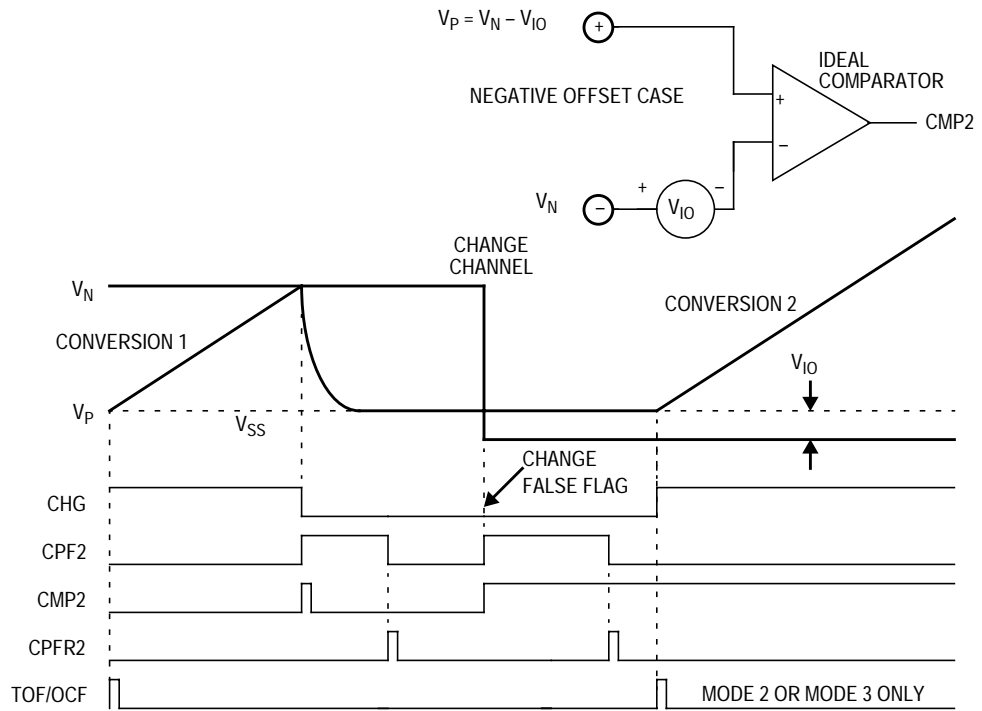
Figure 21. Infinite Conversion Time A/D Fault

Disabled Conversion A/D Fault

The CPF2 flag bit is set by a low-to-high transition on the output of comparator 2. If the output of comparator 2 cannot return to the low state, this flag bit cannot get set again. This occurs whenever the level on the PB0 pin cannot be discharged below the level on the negative input to comparator 2 as in the following two situations (refer to [Figure 7](#)):

- The comparator has a positive input offset voltage *and* the negative comparator input is more than  $V_{IO}$  below  $V_{SS}$ .
- The comparator has a negative input offset voltage *and* the negative comparator input is less than  $V_{IO}$  or below  $V_{SS}$ .

In both these cases, the discharge device cannot pull PB0 below the level on the negative comparator input, and the output of comparator never goes low. After once clearing the CPF2 flag bit, it cannot get set again, and it appears like the infinite conversion fault mentioned before. However, in this case, the ramp voltage does indeed reach the input level, but the CPF2 flag is not set because conversion did not discharge PB0 low enough to make CMP2 go low again. This is referred to also as the “stuck at zero” condition. This is shown in [Figure 22](#) for the negative input offset case.



**Figure 22. Disabled Conversion A/D Fault**

**NOTE:** The CPF2 flag bit may get set incorrectly when changing channels to an input source voltage which is near  $V_{SS}$  as shown in [Figure 22](#). A/D conversion software should account for this case and clear out any unwanted CPF2 flags before starting another conversion.

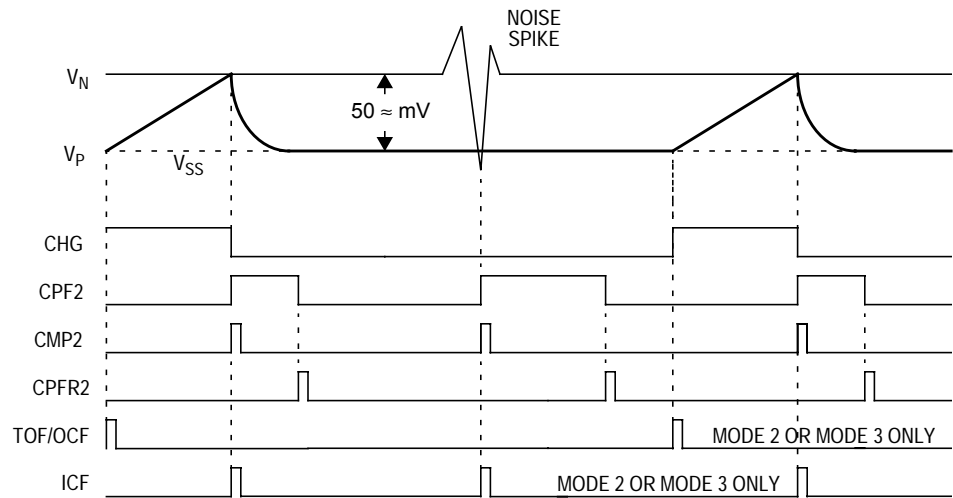
This fault is most likely to occur if the input voltage being converted is within 50 mV of  $V_{SS}$ . The effect of disabled conversion fault can be reduced if the comparator's input voltage is positive as defined by [Figure 7](#). Typical MC69HC05JJ/JP series MCUs tend to have the positive input offset voltage case, with a negative input offset voltage case when the INV bit is set. But, the user should not depend on this always being true, since input offset is a production variation. A software check using the internal  $V_{SS}$  channel could be used to determine which state of the INV bit to use to select the more desired positive offset case discussed in [Comparator Inversion](#) on page 16.

If it does occur, this fault can be detected by the same software check for the maximum conversion time used to detect the infinite conversion

time fault. However, when the maximum time is reached, the CMP2 flag will now be high indicating that the ramp was always above the input level. Therefore, the input is at or below the minimum conversion level, the conversion can be terminated, and the conversion result can be forced to its minimum value.

**Extra Conversion  
A/D Fault**

When the PB0 pin has been discharged and the input voltage is near  $V_{SS}$ , the presence of noise on either the measured input voltage or the PB0 pin can cause comparator 2 to switch as shown in **Figure 23**, and the CPF2 flag bit will be set even though there is no conversion ramp in progress. In this case the polarity of the comparator's input offset voltage is not as important, as this behavior can be seen in both cases.



**Figure 23. Noise on Signals Near  $V_{SS}$**

In mode 0 or mode 1, the user has complete control over the conversion process and can reset any noise induced CPF2 static flags. But, in mode 2 or mode 3, the presence of a noise pulse will not only set the CPF2 static flag bit, but it will trigger an input capture to the 16-bit timer. These captured times appear like completed conversions with unexpected timing. Therefore, the A/D conversion software should clear out any extra CPF2 flags before starting another conversion.

Application Note

Early Conversion  
A/D Fault

When the conversion time is long, the possibility exists for noise on either the input signal or the PB0 ramping voltage to cause comparator 2 to trip the CPF2 flag earlier than would normally have occurred as shown in **Figure 24**. In this case, the polarity of the comparator's input offset voltage is not as important, as the behavior can be seen in both cases.

This problem is aggravated by long, slow-charging ramps where the input signal may be near the ramping voltage for a longer period of time. In general, it is best to use the fastest conversion time possible to reduce the effects of noise on the input signal.

Another cause of an early conversion is to have some charge left on the ramp capacitor at the start of the conversion. Always allocate enough time to allow the ramp cap to fully discharge to  $V_{SS}$ . A good rule of thumb is to allow at least 1/10 the charge time to discharge the ramp cap. If the software process that is running between the start of each A/D conversion is always long enough, then there does not need to be a specific discharge time delay written into the software.

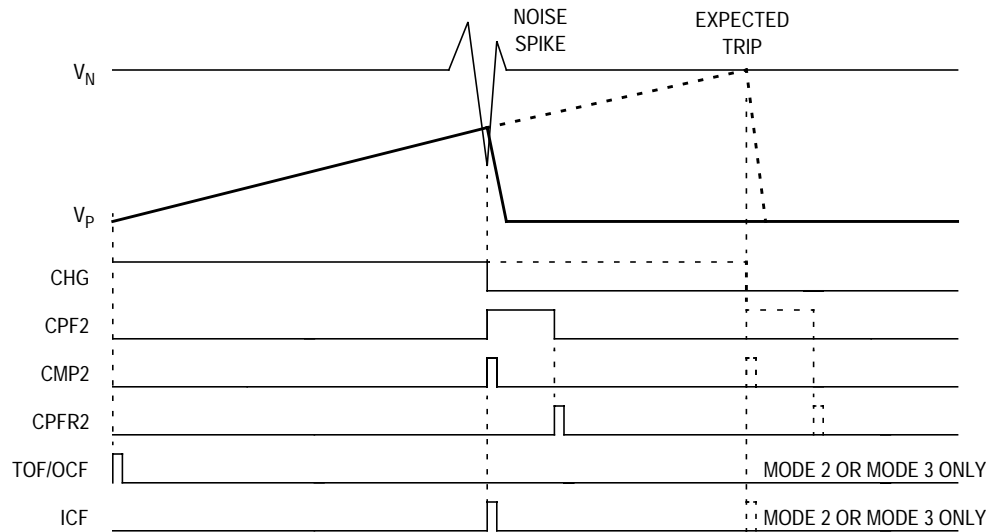


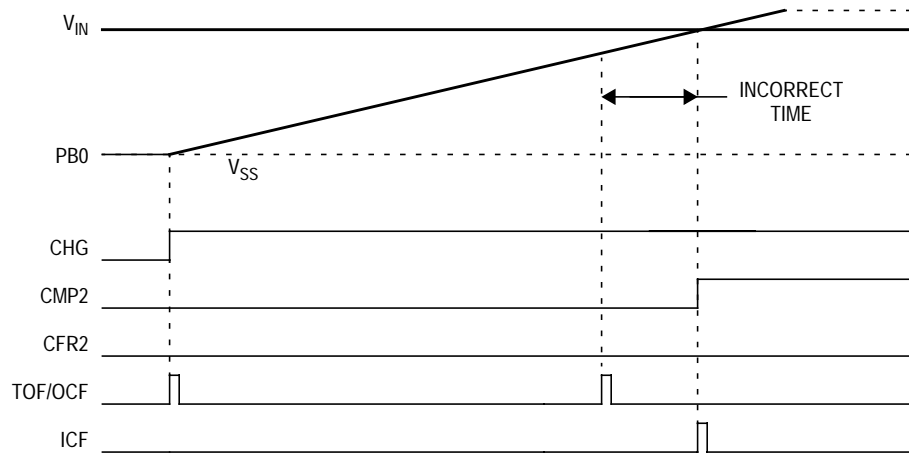
Figure 24. Early Conversion A/D Fault



Early conversions are difficult to detect unless the input signal is slow changing allowing several readings to be taken and the average and/or peak value used to determine the true conversion time.

**Conversion  
Overlap A/D Fault**

When the conversion time becomes too long in mode 0 or mode 1, the software is still looping on the conversion and no subsequent events can occur. In mode 2 or mode 3, there exists the possibility that a very long conversion will overlap the next TOF or OCF which is trying to start another conversion. The net result will be that the ICF for the first conversion may appear like a short conversion associated with the start of the second conversion as shown in **Figure 25** or the first conversion may look like the infinite conversion fault mentioned above. This fault is only associated with mode 2 or mode 3.



**Figure 25. Conversion Overlap A/D Fault**

The problem of conversion overlaps can be detected if one or more software flags are present so that the TOF or OCF interrupt routine can determine that a prior conversion has been started but not finished.

Two suggested software flags are:

- CIP — Conversion in progress
  - Set when TOF or OCF occurs starting charge cycle
  - Clear when ICF detected at end of conversion

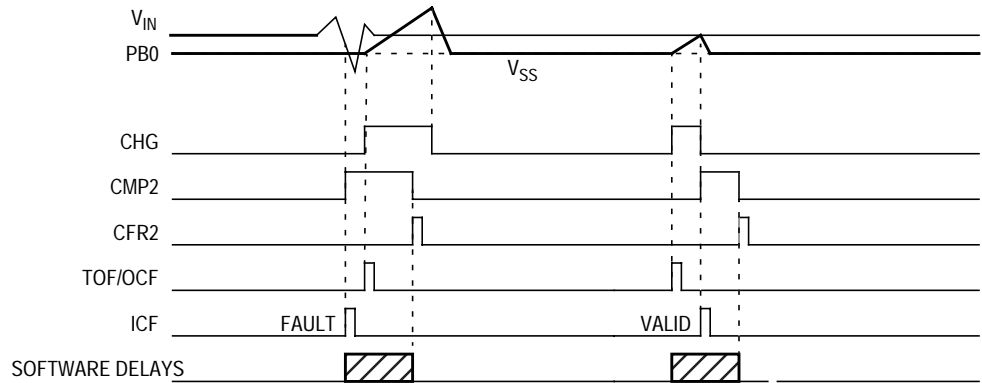
- EOC — End of conversion
  - Set when ICF detected at end of conversion
  - Clear when TOF or OCF occurs starting charge cycle

It may appear that a single software flag is needed to determine if the conversion is running or complete. But the two flags also allow for an initialization case (both clear) or a faulty sequence (both set). With the use of the CIP and EOC flags, the conversion overlap fault can be detected when the next TOC or OCF occurs. If the EOC flag is clear and the CIP flag is still set, then there is an overlap and the next ICF is used for the conversion or ignored. If used, the time should account for any timer roll over that may have occurred. If not used, the integration ramp should be terminated (by manually clearing the CHG bit) and any ICF ignored (by reading the LSB of the input capture registers). In either case, there will be one conversion cycle dropped to use or clear out the overlapping ICF.

#### Conversion Masking A/D Fault

In mode 0 or mode 1, the software handles all the events of the conversion in an established subsequence such that simultaneous events do not happen. In mode 2 or mode 3, there exists the possibility that an unwanted conversion fault may make an ICF occur near the same time as the TOF or OCF, which is trying to start another conversion as shown in [Figure 26](#). This ICF cannot be evaluated until after the software latency getting into the timer interrupt service routine and any software processing before the actual test of the ICF bit. The problem is that such an unwanted ICF just before the TOF cannot be discerned from a true ICF that would occur if the input voltage were near  $V_{SS}$ . This fault is only associated with mode 2 or mode 3.

This fault can be detected by looking at the captured time for the ICF. If it is before the TOF or OCF event (a negative number), then it was a fault and should be ignored, the CPF2 flag cleared, and the input capture released by reading the LSB. If the time difference is a small positive number, then the ICF is assumed to be a valid conversion of a small voltage near  $V_{SS}$ .



**Figure 26. Conversion Masking A/D Fault**

## Manual A/D Conversions

Mode 0 or mode 1 conversions are completely under the control of the software making all events predictable and occurring in a known sequence.

The main limitation of mode 0 and mode 1 is that the timing will be slower either due to software timing loops or slower accesses to the 16-bit timer or multifunction timer. Therefore, mode 0 or mode 1 is usually limited to 8-bit conversions unless extremely long conversion times can be tolerated in the application.

### Software Loop Timing

Timing the A/D conversion in software timing loops requires careful attention to loop times and cycle times between the events that start charging ramp and test the output of comparator 2. Also, all interrupts must be suspended while the conversion is in progress, since preserving accurate software loop times is important. In applications which require rapid interrupt, servicing of events may not work well with mode 0 or mode 1 if software timing loops are used.

### Timing Via Timer Reads

Timing of the conversion can be done by reading the state of the 16-bit timer or the multifunction timer. The software must account for the delays in reading these registers and the asynchronous clocking of these

timers with respect to the instruction cycles. These sources of timing “jitter” will degrade the accuracy of the readings.

#### Analog Interrupt Timing

The conversion can be timed with a combination of software start of the charging ramp and an analog interrupt which reads the 16-bit timer or multifunction timer. This mode of operation will allow other tasks to be performed while the conversion is in progress, which is especially useful for long conversions. However, the accuracy of the analog interrupt in capturing the ending time means that all other interrupts may not be active until after the conversion is complete.

Also, once the analog interrupt is entered, the time that is read must be corrected for the minimum interrupt latency time to get into the analog interrupt service routine and read the timer state. This latency, and, therefore, the accuracy of the time, will be affected by the variation in cycle times of all instruction types used in the particular application software. Avoiding long cycle time instructions such as JSR, SWI, MUL, and some indexed addressing modes can reduce this variation from nine to three bus cycles.

#### Instruction Cycle Timing

The user should be aware of the exact cycle when hardware changes and data reads occur when using the manual A/D conversion techniques. The accuracy of the software timing and the “rounding” of the time count will depend on knowing the precise cycle and clock edge when the charge time begins and when the voltage comparator output is sampled. Two examples of simple mode 0 flows are shown in [Figure 27](#).

In the flow for ATDGO1, a simple loop checks for the CPF2 flag and increments the accumulator. The flow for ATDGO2 includes some added delays to position the first check of the CPF2 flag to be one-half the number of the bus cycles that occur in each loop that follows. This places the CPF2 flag check at one-half of a “bit time” for proper rounding of partial bits. The details of these two examples are given in the two following software examples, and the timing is shown in [Figure 28](#).

Refer to the application note *A/D Conversion Software for the MC68HC05JJ/JP Series Microcontrollers*, Freescale document order number AN1739/D, for detailed software examples on how to implement a mode 0 or mode 1 A/D conversion.

Also, refer to the application note *Instruction Cycle Timing of MC68HC05JJ/JP Series Microcontrollers*, Freescale document order number AN1738/D, for additional information on the exact cycle when hardware changes and/or reads occur in the JJ/JP-series microcontrollers. This application note also shows how to measure these timings in other MC68HC05 Family members.

Application Note

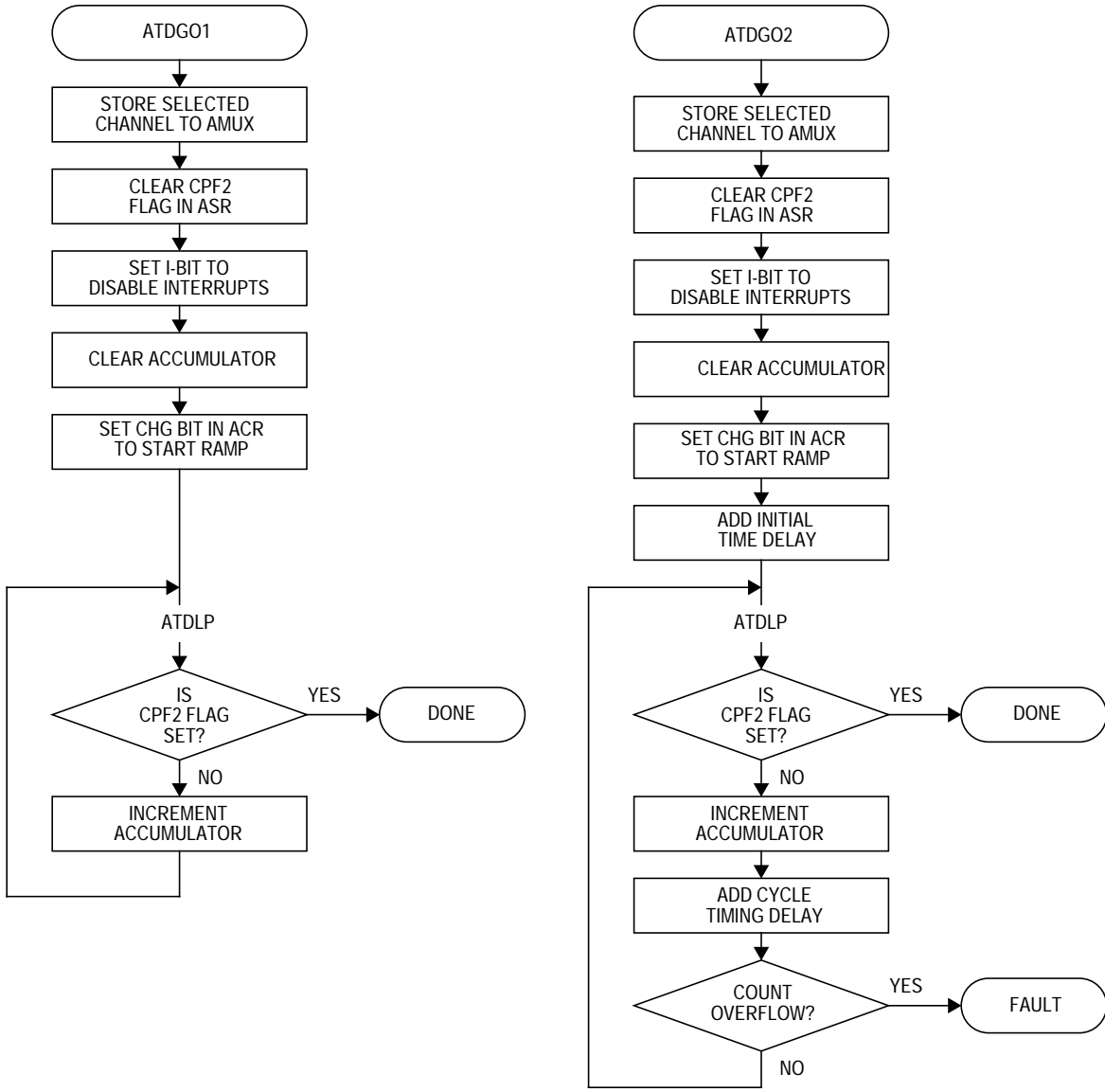


Figure 27. Mode 0 Timing Flowchart

Freescale Semiconductor, Inc.

**Software**
**Example 1 —  
Simple Mode 0**
**A/D Timing**

```

1 0800                                org      $0800
2 0800 1A1E [ 5] ATDGO1 bset      CPFR2,ASR      ;Set CPFR2 to clear CPF2
3 0802 9B   [ 2]         sei                          ;Hold off interrupts
4 0803 4F   [ 3]         clra                         ;Clear accumulator count
5                                         ; to reset conversion count
6 0804 1E1D [ 5]         bset      CHG,ACR          ;Fire off current source
7                                         ; ramp. Actual write to CHG
8                                         ; occurs one cycle before
10                                        ; next opcode is fetched.
11 0806 0E1E03 [ 5] ATDLP brset    CPF2,ASR,DONE   ;Test state of comp 2 flag
12 0809 4C   [ 3]         inca                         ;Increment ACCA.
13 080A 20FA [ 3]         bra       ATDLP           ;Branch back to flag check.
14 080C 9D   [ 2] DONE   *
    
```

**Software**
**Example 2 —  
Balanced Mode 0**
**A/D Timing**

```

1 0800                                org      $0800
2 0800 1A1E [ 5] ATDGO2 bset      CPFR2,ASR      ;Set CPFR2 to clear CPF2
3 0802 9B   [ 2]         sei                          ;Hold off interrupts
4 0803 4F   [ 3]         clra                         ; to reset conversion count
6 0804 1E1D [ 5]         bset      CHG,ACR          ;Fire off current source
7                                         ; ramp. Actual write to CHG
8                                         ; occurs one cycle before
9                                         ; next opcode is fetched.
10 0806 21FE [ 3]         brn      *                ;Add 3 cycle delay, using
11                                         ; BRN with dummy branch
12                                         ; to same instruction.
13 0808 0E1E07 [ 5] ATDLP brset    CPF2,ASR,DONE   ;Test state of comp 2 flag
14                                         ; at 1/2 of the software
15                                         ; counter's bit time.
16                                         ;Carry remains clear if
17                                         ; CPF2 tests as being clear.
18 080B AB01 [ 2]         add      #$01            ;Increment ACCA. Carry is
19                                         ; cleared unless there is
20                                         ; a carry from the add
21                                         ; operation itself
22 080D 9D   [ 2]         nop                          ;Add 4 cycle delay, using
23 080E 9D   [ 2]         nop                          ; (2) NOP instructions
24 080F 24F7 [ 3]         bcc      ATDLP           ;If accum. count reaches
25                                         ; maximum, then carry will
26                                         ; be set. If carry is clear
27                                         ; then A/D loop count should
28                                         ; continue, else continue on
29                                         ; to process fault, with
30                                         ; carry bit set and ACCA
31                                         ; left equal to $00.
32 0811 9D   [ 2] FAULT  *                          ;Fix fault, over time limit
33 0812 9D   [ 2] DONE   *
    
```

AN1740

Application Note

Freescale Semiconductor, Inc.

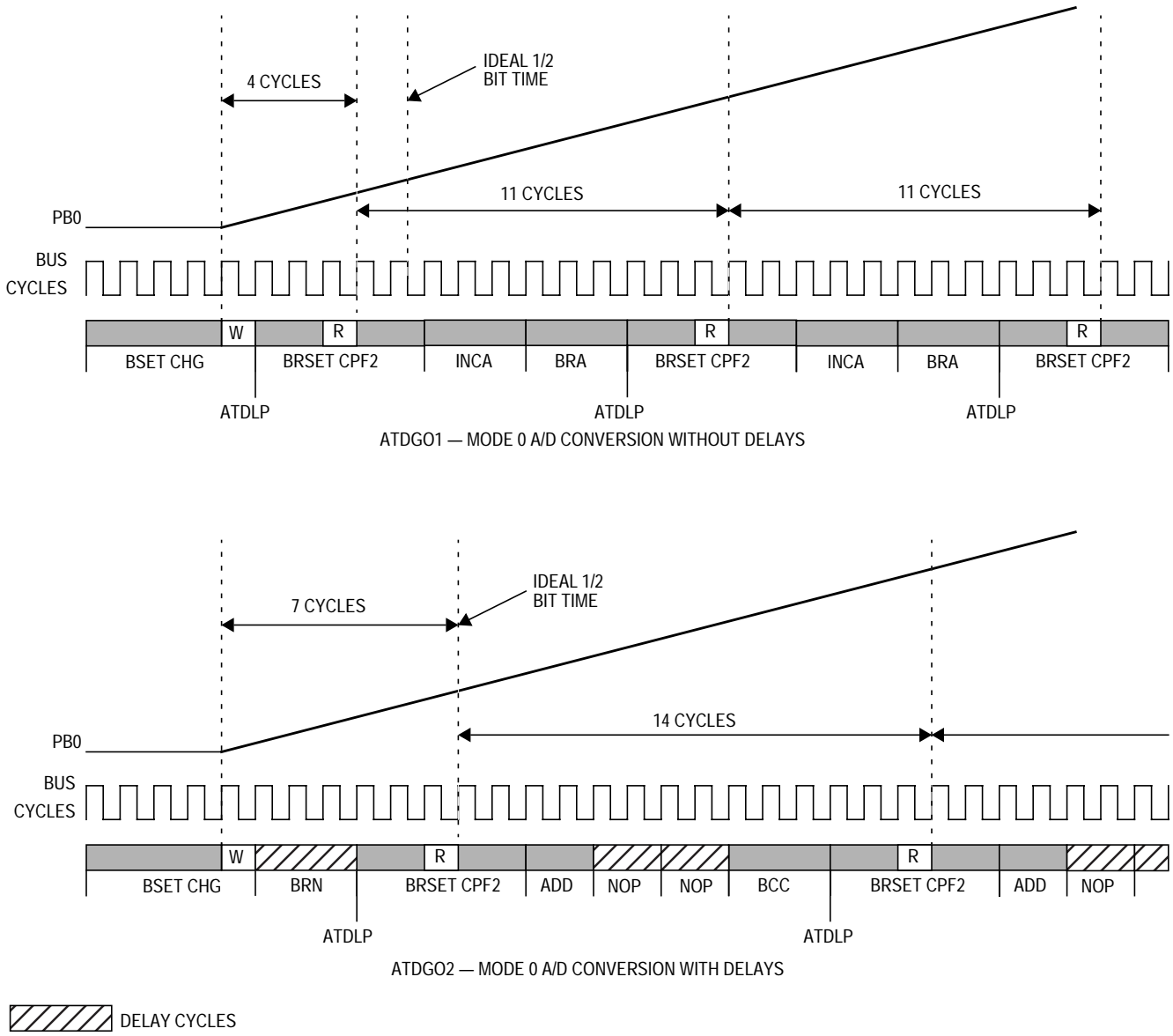


Figure 28. Balancing Conversion Startup and Cycle Times



## Automated A/D Conversions

---

Mode 2 and mode 3 conversions are mostly under the control of the 16-bit timer, but they need occasional maintenance to keep flags cleared and to transfer input capture times to RAM for use by other routines. Due to the more accurate start and stop timing and high possible bit counts, mode 2 or mode 3 conversions are best for long conversions of more than eight bits.

Mode 2 or mode 3 conversions use the timer overflow flag (TOF) or the output compare flag (OCF) and the input capture flag (ICF) of the 16-bit timer to control the charging of the integration capacitor connected to PB0. When using these timer flags, the designer should be aware of these behaviors:

1. The TOF flag bit is set whenever the 16-bit timer counter overflows from \$FFFF to \$0000 *regardless of whether or not the timer overflow interrupt is enabled* (TOIE bit set in TCR).
  - a. Once set, the TOF bit must be reset by the proper software sequence of reading the TSR, then the TMRL.
  - b. Since the timer starts with \$FFFC, a TOF will occur almost immediately after a reset.
2. The OCF flag bit is set whenever the 16-bit timer counter matches the count stored into the output compare registers *regardless of whether or not the timer output compare interrupt is enabled* (OCIE bit set in TCR).
  - a. Once set, the OCF bit must be reset by the proper software sequence of reading the TSR, then the OCRL.
  - b. Once set, the OCF will lock out any future compares until the OCF flag bit is cleared.
  - c. An OCF can occur at any time after a reset, since the contents of the OCRL and OCRH are unaffected by a reset.
3. The ICF flag bit is set whenever an input capture is received by the 16-bit timer *regardless of whether or not the timer input capture interrupt is enabled* (ICIE bit set in TCR).

- a. Once set, the ICF bit must be reset by the proper software sequence of reading the TSR, then the ICRL.
- b. Once set, the ICF will lock out any future input captures until the ICF flag bit is cleared.
- c. The ICF flag overrides the TOF or OCF flags. If a TOF or OCF is present when the ICF is cleared, the charging of the PB0 pin will begin immediately.
- d. The ICF can be triggered by a positive or negative going edge depending on the state of the IEDG bit in the TCR.
- e. The trigger source for the ICF can be either the signal on PB3 or the output of voltage comparator 2 dependent on the state of the ICEN bit in the ACR.

The most important considerations in using the automated mode 2 or mode 3 is that the TOF, OCF, and ICF flags can get set without any enabling control and that they can start getting set *before* the software is ready to use them. This means that modes 2 or 3 can operate as either interrupt-driven or free-running software.

#### Use of Conversion Flags

It is highly recommended that modes 2 and 3 use conversion status flags such as EOC and CIP as described on page 57 and 58. These flags allow the timer interrupt service routine to determine if the conversions are executing properly or if a fault has occurred. These flags should be contained in an assigned RAM location. Avoid using the CHG bit in the ACR or the CPF2 or CMP2 bits in the ASR for these purposes, since they can be set automatically by hardware events that are not under the user's direct software control.

**Value of the  
Captured Times**

The biggest advantage of modes 2 and 3 is that the start time of the conversion is clearly defined by the TOF of OCF timer state and the end of conversion is captured precisely by the ICF. Therefore, the software sequence and cycle times are not critical to the timing accuracy. The only concern should be that the ICF is combined with the correct TOF of OCF that started the conversion ramp. Also, the software must access the proper values and clear the flags before the next conversion is to begin. This allows more latitude in defining error checks than is possible with modes 0 or 1, which are sensitive to software cycle timing.

**Correct Mode  
Initialization**

The initialization of either mode 2 or 3 must be done at the proper time, since the TOF or OCF flags may become set before the mode can be established. If the TOF is already set and mode 2 is initiated, then the charge ramp will begin immediately without regard for when the TOF actually occurred. Similarly, the OCF could be set before mode 3 is initialized. It is important that the TOF and ICF flags be clear before mode 2 is established or that the OCF and ICF flags be clear before mode 3 is established. The best way to accomplish this is to wait for the first TOF in mode 2 or first OCF in mode 3, then clear all flags and finally set the ATD1 and ATD2 bits as required.

Refer to application note *A/D Conversion Software for the MC68HC05JJ/JP Series Microcontrollers*, Freescale document order number AN1739/D, for detailed software examples on how to implement a mode 2 or mode 3 A/D conversion.

## General A/D Techniques

---

The following sections cover some considerations in using channel selection, sample and hold, and other A/D-related topics.

### Channel Selection Techniques

Normally, there is a need to select more than one channel in performing an A/D conversion. This is needed to select between reference and unknown signals as well as to have more than one external voltage to monitor. In the case of modes 0 or 1, it is quite easy to schedule the sequence of events needed to change channels and then execute a conversion since all the steps are under direct software control and timing. But, in the case of modes 2 or 3, the A/D conversion is driven automatically by the 16-bit timer with little interface to the software. In modes 2 or 3, the timing of the channel selection needs additional consideration.

This can be solved easily if we make the channel selection occur at the end of the prior conversion cycle. In both modes 2 and 3, the change in channel would then occur in the service routine for the ICF. If the channel selection is a recurring sequence, then the ICF service routine can merely step to the next channel in the sequence. If, however, there is a need to select the next channel based on changing criteria, then the next channel to be selected can be done with a pointer location in RAM and a pair of software flag bits to handshake between the interrupt-driven ICF service routine and the asynchronously running background software that determines which channel to convert next.

### Sample and Hold Techniques

A sample and hold method can be applied to all A/D conversions. As with making channel selections, sample and hold in modes 0 and 1 can be done easily, while in modes 2 or 3 the timing of the sample needs additional consideration. This can be solved easily, if we consider that the sample can actually last into the start of the conversion cycle.

In mode 2, the triggering event is the TOF. If the OCF is not being used for some other task, it can be used to trigger the start of the sample just before the next TOF occurs. Then in the TOF service routine, the sample

can be terminated. This will make the sample start some time before the A/D conversion start and end a short time after the A/D conversion start.

In mode 3, the OCF cannot be used easily to get a sample start before the A/D conversion begins, since the OCF itself is the starting event. But, the input to be sampled can be selected after the prior A/D conversion ends (when the ICF is serviced) and then terminate the sample shortly after the start of the A/D conversion when servicing the OCF.

**Using the A/D Conversion Results**

Once a time count is determined for a given input signal, it can be compared to a reference count for the full scale voltage,  $V_{REF}$ , and a reference count for the lowest voltage,  $V_{OFF}$ , as follows:

$$V_X = (V_{REF} + V_{OFF}) \times \frac{N_X - N_{OFF}}{N_{REF} - N_{OFF}}$$

where:

- $V_X$  = Calculated value of measured voltage (volts)
- $V_{REF}$  = Value of known reference voltage (volts)
- $V_{OFF}$  = Value of lowest voltage (volts)
- $N_X$  = Count for unknown voltage (counts)
- $N_{OFF}$  = Count for lowest reference voltage (counts)
- $N_{REF}$  = Count for highest reference voltage (counts)

This calculation also is shown as a flowchart for both absolute and ratiometric values in [Figure 29](#).

Application Note

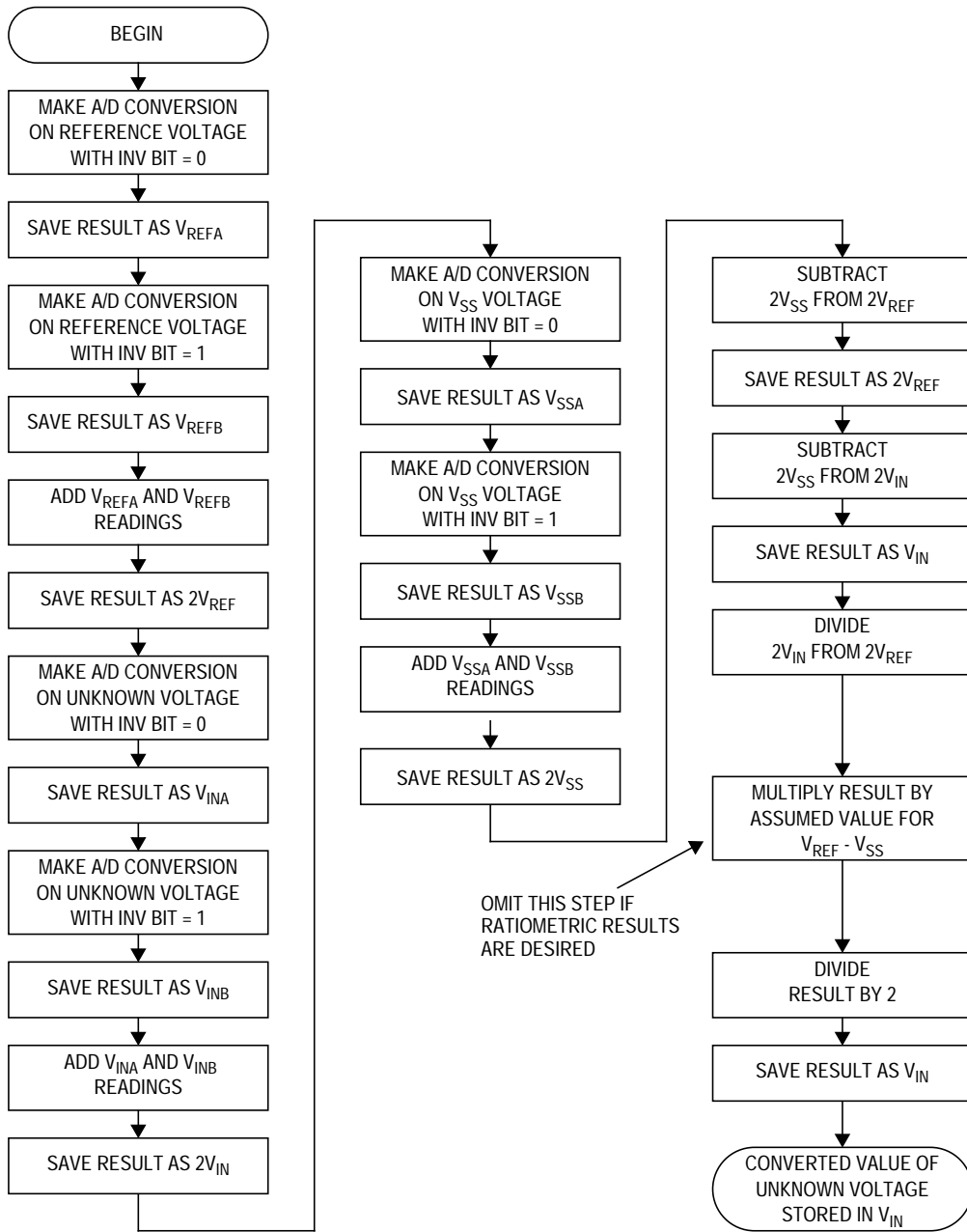


Figure 29. Flowchart for Calculating A/D Results

**Recommended Ramp Capacitors**

The desired capacitor for the ramp capacitor is any of the “poly” film types which have both low leakage and low dielectric absorption (sometimes referred to as memory behavior). Capacitor parameters other than capacitance and working voltage are equivalent series resistance (ESR), equivalent series inductance (ESL), leakage, and dielectric absorption (expressed as RDA or CDA).

Dielectric absorption may leave some charge in the capacitor after a discharge. Such remaining charge will affect subsequent conversion ramps. Leakage acts as a parallel resistor which can make the charging ramp non-linear.

Polystyrene capacitors are usually the best choice. Low-cost monolithic ceramics are good for bypass use, but have high dielectric absorption which makes them less desirable for an integration or storage application. Tantalum or aluminum electrolytics have high dielectric absorption and too much leakage as well. Integration or storage capacitors should have a dielectric absorption of less than 0.01%.

**Internal Analog  $V_{SS}$  Offset**

The JJ/JP series of MCUs has only a single  $V_{SS}$  pin through which flows both the analog and digital currents. Also, any external load which sinks current into the device will pass this current through the  $V_{SS}$  pin. The internal analog and digital  $V_{SS}$  lines are connected to the  $V_{SS}$  pin through separate bonding wires. But the current of both must flow through the leadframe and out of the  $V_{SS}$  pin to the user’s PWB. This combined current can result in a small IR voltage drop which adds a positive offset to the internal analog  $V_{SS}$ , called  $V_{A\text{OFF}}$  in **Figure 1**. The resistance of the  $V_{SS}$  pin is about 0.1 ohm, which means that internal analog  $V_{SS}$  will rise about one millivolt for every 10 milliamperes of current that flows through the  $V_{SS}$  pin.

**Current Spikes**

Similar to the voltage offset just described, current spikes from high-speed digital switching circuits will generate a voltage drop across the package pin shared with the analog subsystem. The voltage drop due to the resistance of a package lead is minor compared to the inductance of the lead, which is particularly a problem with fast switching currents.

Current spikes can occur when discharging an external capacitor through PB0 which can be disruptive to the internal analog  $V_{SS}$  with respect to the external  $V_{SS}$  pin.

**NOTE:** *Discharging large capacitors or low impedance loads into the PB0 pin can cause high current spikes which may disrupt the internal analog  $V_{SS}$  with respect to the  $V_{SS}$  pin and cause unwanted analog noise. Repeated high currents can also damage the discharge device and/or the device's interconnect metal. Using capacitors that are less than  $2\mu\text{F}$  and discharge current peaks less than 25 mA is recommended.*

These current transients cause a voltage to the analog subsystem  $V_{SS}$  reference to rise or drop several tenths of volts during switching transients. Therefore, these recommendations are given:

1. Avoid switching any port pins which carry more than a few milliamperes of current while doing A/D conversions.
2. Sample the output of the comparators between the switching of port conditions and use the dynamic outputs on CMP1 and CMP2.
3. If the CPF1 and CPF2 static flags are needed, be sure to clear out any flags which may have been set during switching spikes by including a flag clearing sequence just after the code which initiated the suspected current spike.
4. Try to position output changes within the same time in each cycle of operation and for each measurement. Transients with low frequency components, therefore, always will affect all readings in a similar manner.
5. Consider making the A/D conversions during wait mode when the internal MCU bus activity has been halted.

## System Noise

System noise is inherent to a mixed signal design such as the JJ/JP series of MCUs. There will always be some level of noise which cannot be filtered out, bypassed, or avoided by careful PWB layout. The single slope A/D method ensures some level of averaging of the input signals, but additional digital averaging may be necessary to reduce the unwanted noise further. The penalty in the added filtering will be a reduction in response time to the input signal.



Refer to the application note *In-Circuit and Emulation Consideration for MC68HC05JJ/JP Series Microcontrollers*, Freescale document order number AN1741/D, for more information on good PWB layout techniques and limiting these noise sources.

## Analog Subsystem Low-Power Modes

The JJ/JP series of MCUs has the low-power modes common to other Freescale MC68HC05 devices. CMOS digital logic consumes very little power when no switching is occurring. Wait and stop modes specifically reduce the number of digital functions which can switch. However, analog circuits continue to consume power as long as they are connected to a power source. Therefore, the amount of supply current required in wait and stop modes highly depends on the use of the analog subsystem. Similar to the 16-bit timer, the analog functions will remain active during wait mode. Therefore, these comparators can be used to wake up the device from wait mode by the analog interrupt.

The analog functions can also be left active during stop mode, but the user must realize that a higher stop  $I_{DD}$  current will exist. Tripping the comparator inputs will not restart the device from stop mode, but the output of comparator 1 can be routed out through the PB4 pin by setting the COE1 bit in the ASR. In this case, PB4 can be connected to the  $\overline{IRQ}$  or  $\overline{RESET}$  pin to initiate a restart if the output of comparator 1 goes low.

### Use During Wait Mode

In wait mode, the CPU bus activity is stopped, but the clocks to the two timer functions are kept active. The analog functions do not require clocks to operate, so the comparators and the current source will still be active in wait mode if they are not powered down prior to executing the wait instruction. An advantage of wait mode is the reduction of digital switching noise as previously discussed. This makes wait mode a good candidate for taking “low noise” readings with the analog subsystem.

Also in wait mode, comparator 1 can have its output directed to the PB4 pin using the OPT mask option and the COE1 bit in the ASR. In this case, comparator 1 can sense a condition on its inputs and direct some external action without waking up the CPU.

**NOTE:** *Comparator 1 can be used to control an external signal using the COE1 function, even if the device is in wait mode.*

#### Use During Stop Mode

The comparators can remain active during stop mode, but cannot directly restart the part unless the output of comparator 1 is directed externally to PB4 using the OPT mask option and the COE1 bit in the ASR. Then PB4 is connected externally to an  $\overline{\text{IRQ}}$  pin or the  $\overline{\text{RESET}}$  pin.

**NOTE:** *Comparator 1 can be used to control an external signal using the COE1 function, even if the device is in stop mode.*

## Design Check List

**Table 7** contains a listing of major considerations for each feature of the analog subsystem with a reference to the page of the section where the item is discussed. Many of these topics are given earlier as notes.

**Table 7. Design Checklist**

Analog subsystem feature	Design consideration	Refer to page number
Comparators	At least one input must always be within the common mode range.	<a href="#">14</a>
Comparators	The INV bit should be used when one input is within $V_{IO}$ of $V_{SS}$ .	<a href="#">17</a>
Comparators	Always clear the CPF1 and CPF2 ag bits after changing the INV bit.	<a href="#">19</a>
Comparators	Input pulse times should be longer than the sample rate of CMP1 or CMP2.	<a href="#">21</a>
Comparators	Analyze code used with comparators for exact read and write cycle timing.	<a href="#">63</a>
Comparators	Should input pulldowns on port B be OFF when using comparators?	<a href="#">42</a>
Comparators	Allow time for the comparators and the bias source to power up <i>before</i> sampling their outputs or trying to clear any CPF1 or CPF2 ags .	<a href="#">21, 45</a>
Current source	Clear the ISEN to disable the current source and reduce power consumption	<a href="#">25</a>
Discharge device	Avoid capacitors greater than 2 $\mu$ F and discharge currents more than 25 mA.	<a href="#">28, 72</a>
Channel multiplexers	The channel multiplexers require that any make-before-break or break-before-make considerations are done in software with separate write cycles to the AMUX register.	<a href="#">29</a>
Channel multiplexers	The input divider ratio will be altered if multiple channels are turned on in parallel at the same time.	<a href="#">32, 37</a>
All A/D modes	Be sure to turn off the pulldown on PB0 when making A/D conversions. If this pulldown is left on the resulting charge characteristic will look more like an RC charge curve than a constant current ramp.	<a href="#">42</a>
All A/D modes	Make allowance for the worst case variation in the external capacitor, oscillator frequency, full scale voltage and charging current when planning maximum and minimum time requirements.	<a href="#">50</a>
All A/D modes	Provide checks and corrections for the six faults that can occur with single slope A/D converters.	<a href="#">52</a>
All A/D modes	Use a proper type of capacitor for the external ramp capacitor to avoid problems with linearity on long conversion times.	<a href="#">71</a>

**Table 7. Design Checklist (Continued)**

Analog subsystem feature	Design consideration	Refer to page number
Mode 0 A/D Mode 1 A/D	Check software timing to be sure that the first check of the comparator output occurs at 1/2 of the bit time (loop time between future checks).	<a href="#">63</a>
Mode 2 A/D	Make sure that the OCRL is read after each read of the OCRH. Otherwise the read of the OCRH will hold off future output compares until the OCRL is read.	<a href="#">65</a>
Mode 2 A/D Mode 3 A/D	Avoid changes in ICEN, ICIE, or IEDG during charging of the external capacitor, as these may cause unwanted ICF flags to be generated.	<a href="#">44</a>
Mode 2 A/D Mode 3 A/D	In order to clear the ICF, OCF or TOF, the TSR must be read first; and then the respective LSB of each related timer register must be read.	<a href="#">65</a>
Mode 2 A/D Mode 3 A/D	Presence of the ICF flag will override both the OCF and TOF flags. Must clear OCF or TOF before clearing the ICF or another charge cycle will start immediately when the ICF is cleared.	<a href="#">44</a> , <a href="#">66</a>
Mode 2 A/D Mode 3 A/D	Avoid using both the analog interrupt and the input capture interrupt for detecting a change in the output of comparator 2.	<a href="#">24</a>
Sample & hold	Do not change the state of the INV or VOFF bits while the voltage sample is being held on the sample capacitor.	<a href="#">38</a>
Sample & hold	Connect the channel selection bus to V <sub>REF</sub> <u>after</u> the HOLD and DHOLD bits have been turned off.	<a href="#">39</a>
Sample & hold	Avoid using the sample and hold method in severe noise environments as the noise can corrupt the sampled voltage during the hold time.	<a href="#">39</a>
Sample & hold	Avoid switching high currents with the port pins during the hold time.	<a href="#">38</a> <a href="#">71-72</a>
Sample & hold	Be sure to turn off the HOLD or DHOLD bits <u>before</u> making any changes to the channel selection. Turn off the HOLD or DHOLD on a separate write cycle to the AMUX register.	<a href="#">40</a>
All analog features	Be sure that the window on erasable EPROM devices is covered with an opaque material such as black electrical tape. Light will affect currents, trip points, and leakage in the analog subsystem.	<a href="#">38</a>
All analog pins	Be sure that the pulldowns are turned OFF when using PB0:4 for analog signal processing.	<a href="#">42</a>
Temperature diode	Make sure that the analog bias is on when measuring the temperature diode. This can be done by setting ISEN, CP1E or CP2E in the ACR.	<a href="#">41</a>
Optional features	Make sure that the OPT bit in the MOR (EPROM devices) or the correct mask option is selected when either the VOFF or COE1 functions are required.	<a href="#">4</a> , <a href="#">6</a> , <a href="#">22</a> , <a href="#">40</a>

## References

---

*Instruction Cycle Timing of MC68HC05JJ/JP Series Microcontrollers*, Freescale Application Note AN1738, 1998.

*A/D Conversion Software for the MC68HC05JJ/JP Series Microcontrollers*, Freescale Application Note AN1739, 1998.

*In-Circuit and Emulation Considerations for MC68HC05JJ/JP Series Microcontrollers*, Freescale Application Note AN1741, 1998

*Reducing A/D Errors in Microcontroller Applications*, Freescale Application Note AN1058, 1990.

*CMOS Analog Circuit Design*, Phillip E. Allen and Douglas R. Holberg, Oxford University Press, New York, N.Y., 1987, ISBN 0-19-510720-9.

Guinta, Steve, "Capacitance and Capacitors," *Analog Dialogue 30-2*, Analog Devices, Inc., 1996.

Kester, Walter, "A Grounding Philosophy for Mixed-Signal Systems," *Electronic Design*, Analog Applications Issue, June 23, 1997.





## 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

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document. Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor 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 which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor 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 Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor 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 Freescale Semiconductor was negligent regarding the design or manufacture of the part.