June, 2007
Design Tips & Tricks for Analog Functions on 8-bit MCUs
AZ309

John Suchyta
8-Bit MCD Applications Engineer
Agenda

► Analog to Digital Converter
► Analog Comparator
► Crystal and PLL
► Current Injection
► Unused Pin Termination
Analog To Digital Converter
Analog To Digital Converter

- Analog to Digital Converters are modules that are re-used among MCU sub-families
- Basic elements of ADC modules
  - Power supply
  - Reference supply
  - Analog inputs
  - Channel selector
  - Analog converter
  - Clock
  - Control
  - Data registers

Simplified ADC Block Diagram
Supply and Reference Pins

► High pin-count MCUs generally have separate ADC power supply and reference pins

► Even so, the VDDAD and VREFH supplies are diode clamped to VDD for circuit protection. Same with VSSAD and VREFL to VSS

► Keep VDD, VDDAD, VREFH at same potential

► Keep VSS, VSSAD, VREFL at same potential

► Tip is to filter ADC supply and reference separately from digital power supply (VDD)
Analog To Digital Converter

Supply Filtering

- **VDD and VSS** bypassed with 100nF close to MCU
- **VDDAD and VREFH** decoupled from VDD with resistor or inductor
- **VSSAD and VREFL** decoupled from VSS with resistor or inductor
- **VDDAD/VREFH** and **VSSAD/VREFL** bypassed with 100nF close to MCU

VDDAD and VREF Filtering
Analog To Digital Converter

Supply and Reference Pins

► Low pin-count MCUs combine supply and reference sources internally with chip VDD and VSS
► This puts more emphasis on appropriate noise filtering on the VDD/VSS power supply
► **Tip** is to use 0.1μF cap very close (within 0.5cm) to chip supply pins
► Additional **tips** include software disciplines to limit effects of noise

Ref: AN2764
Supply and Channel Filtering

- VDD and VSS *bypassed* with 100nF close to MCU
- “Analog” power and ground *isolated* from “digital” power and ground
- **Tip** - place channel filter cap close to MCU
- **Tip** - place channel resistor close to source
- **Tip** - for off-board inputs, physically place resistors in parallel so that noise does not couple to filtered nodes
ADC Input Equivalence

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>CADIN</td>
<td>5 – 15</td>
<td>pF</td>
</tr>
<tr>
<td>RADIN</td>
<td>3 – 15</td>
<td>kΩ</td>
</tr>
<tr>
<td>RAS</td>
<td>Max 10</td>
<td>kΩ</td>
</tr>
<tr>
<td>CAS</td>
<td>0 - 100</td>
<td>nF</td>
</tr>
<tr>
<td>ADC Clock</td>
<td>0.4 – 8.0</td>
<td>MHz</td>
</tr>
<tr>
<td>Sample Time</td>
<td>3.5 – 30</td>
<td>ADC Clocks</td>
</tr>
</tbody>
</table>
ADC Input Equivalence

► The sample path resistance and capacitance are sized so that, if the external source resistance maximum of 10kΩ is met, the accuracy spec’s can be met.

► The overall impedance of the input can be modeled as:
  > External source resistance RAS (specified max 10kΩ)
  > External board/source/pin capacitance CAS
  > Routing and sampling resistance RADIN
  > ADC DAC capacitance CADIN

► Of primary importance for AC signals is ensuring RAS can source CAS in the required time
  • Simple lumped model of (RAS+RADIN)(CAS+CADIN) can be used for noncritical applications
  • Sampling requirement is that system can charge to 1/4th an LSB in sampling window
Simplified ADC Clocking

- Sample time is dependent on MCU and options (3.5 – 30 clocks)
- Conversion time depends on ADC clock, resolution, sample time, and MCU bus rate
- Sampling requirement is that the source can charge the sample cap to ¼ of LSB in sample window.
ADC clock, not bus clock, is timebase

Sample time is dependent on MCU and options
- Low power vs high speed
- Short sample vs long sample

Conversion time depends on ADC clock, resolution, and sample time

Charge to 1/4th LSB means that 8.32 time constants must occur in sample window (10-bit ADC)

<table>
<thead>
<tr>
<th>Sample Window (us)</th>
<th>ADC Clock (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample Time (cycs)</td>
<td>0.50</td>
</tr>
<tr>
<td>3.5</td>
<td>7.00</td>
</tr>
<tr>
<td>4</td>
<td>8.00</td>
</tr>
<tr>
<td>5</td>
<td>10.00</td>
</tr>
<tr>
<td>14</td>
<td>28.00</td>
</tr>
<tr>
<td>24</td>
<td>48.00</td>
</tr>
</tbody>
</table>
RAS and CAS Example

- MCU = MC9S08QG8 (10-bit)
- RADIN = 7kΩ max
- CADIN = 5.5pF max
- Sample time = 3.5 ADC clocks
- ADC Clock = 8, 4, 2, 1, and 0.5MHz

<table>
<thead>
<tr>
<th>ADC Clock (MHz)</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Conv Freq (kHz)</td>
<td>400</td>
<td>200</td>
<td>100</td>
<td>50</td>
<td>25</td>
</tr>
<tr>
<td>Sample Time (us)</td>
<td>0.44</td>
<td>0.88</td>
<td>1.75</td>
<td>3.5</td>
<td>7.0</td>
</tr>
</tbody>
</table>

Assume RAS = 1
Calculate max CAS

| RAS (kΩ) | 1 | 1 | 1 | 1 | 1 |
| CAS (pF) | 1 | 8 | 21 | 47 | 100 |

Assume CAS = 0
Calculate max RAS

| CAS (pF) | 0 | 5 | 10 | 22 | 47 |
| RAS (kΩ) | 2.6 | 3.0 | 6.6 | 8.3 | 9.0 |
RAS and CAS

Implications of RAS = 10kΩ max
- For high speed signals, CAS must be small
  - Run multiple conversions to average for greater accuracy
- For low speed (DC) signals, CAS can be up to 100nF
  - Helps with noise on low speed signal
  - Use longer sample time

Higher values of RAS can be used with corresponding longer sample times
- Newer HC08 and S08 ADC modules have long sample option

CAS is not necessary if signal impedance is low enough or if signal is free from noise
Usage Model

► Port Initialization: Configure all ADC channels as inputs
  • ADC will override port configuration, but not until conversions are started
  • If a pin within a port has an analog voltage present, the input circuit may have a brief (one cycle) surge of current when the port is read. Read the Data Register only when necessary to limit current

► Separate ADC code from other routines if possible
  • For best accuracy, conversions should be performed in WAIT (or STOP)
  • Interrupts from other sources which may occur during conversions could cause missed data conversions in continuous convert mode.
  • If separation is not possible, ensure bus speed and ISR routines are fast enough to ensure no lost conversions in worst case of conflicting interrupt requests

► Operation in WAIT mode
  • Low pin-count MCUs do not have dedicated power supplies for the ADC, so CPU execution will add noise to the supply and lower conversion accuracy
  • Operation in WAIT mode is recommended to lower bus noise
Application Recommendations - Accuracy

► Bypass from VDD to VSS is Required
  • VDDA and VREFH are shared with VDD internally, as are VSSA/VREFL/VSS
  • 0.1μF cap within 0.5cm of package pin; 10μF cap on board MINIMUM
  • If inductors are used, there must be an additional 1μF cap on MCU side of inductor
► Use WAIT immediately following start of conversion
► If in RUN mode, take extra care of board design to reduce noise
► No pin switching – input or output – during conversion
► Use internal clock source (crystal/resonator generates IO noise)
► Run in STOP mode using asynchronous clock (single convert, DC only)
► Average 4 or more results (can reduce one-time noise issues)
► Use the largest capacitor on the input that the sample rate will allow (at least 100pF)
► Use SPI/Timer/etc. interrupt routines to discard ADC results which may have conflicted with IO activity
References

- AN2438 – ADC Definitions and Specifications
- AN2764 – Improving the Transient Immunity Performance of Microcontroller-Based Applications
- AN2321 – Designing for Board Level Electromagnetic Compatibility
- AN1853 – Embedding Microcontrollers in Domestic Refrigeration Appliances
- AN3031 – Temperature Sensor for the HCS08 Microcontroller Family
- AN3266 – Getting Started with RS08
- AN3409 – Basic Refrigerator Control Using the MC9RS08KA2
- DRM079 – Variable Speed DC Fan Control using the MC9RS08KA2
Analog Comparator
Analog Comparator

► ACMP+ – Non-inverting input
  • External input
  • Internal reference
► ACMP- – Inverting input
► ACMPO – Comparator output
  • Internal flag/interrupt
  • Internal timer input
  • External signal
► Output = 1 when V+ > V-
► Output = 0 when V+ < V-
► Not an op-amp

Available on:
MC9S08QG8
MC9RS08KA2
Comparator Example

Battery Voltage Monitor

► Using internal reference
  • Compare to bandgap reference
  • Vbg = 1.2V
  • Vb = 5V
  • R1/R2 = 31k/10k (1.2V)

► Using external reference
  • Vb = 5V
  • VCC = 3.3V
  • R1/R2 = 31k/10k (1.2V)
  • R3/R4 = 8.2k/4.7k (1.2V)
Comparator Example

Hysteresis

- Comparator has small amount of hysteresis
- Noisy inputs can cause multiple output transitions when input level is close to input threshold
- Use output to bias input threshold
- Given VCC=VOhigh=3.3V
- Set Vth=1.9V; Vtl=1.4V
- Pick R1=6.8k
- Solve for R2=6.8k; R3=19k (use 18k)

\[
V_{th} \left( \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} \right) = \frac{V_{CC}}{R_1} + \frac{V_{Ohigh}}{R_3}
\]

\[
V_{tl} \left( \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} \right) = \frac{V_{CC}}{R_1}
\]
Comparator Example

Hysteresis - Results

► Target Vth=1.9V; Vtl=1.4V
► Target hysteresis of 500mV
► Trace 1 is input 3Vp-p sine wave
► Trace 2 is comparator output
► Vth=2.0V
► Vtl=1.4V
► Hysteresis=600mV
Comparator Example

Voltage Doubler

► R4 and C3 provide the frequency for a square wave at the comparator output
► Square wave means Vth and Vtl are centered around VCC/2
► R1, R2, and R3 for hysteresis
► C1 and C2 form the doubler, C1 = C2, along with D1 and D2
► C2 and RL provide output filtering to minimize ripple
► D1 and D2 are high speed diodes (1N4148)
Comparator Example

Voltage Doubler (continued)

► Hysteresis thresholds are 1.9V and 1.4V
► Pick period=1ms, F=1kHz
► Pick C3=0.1uF, solve for R4
  • R4=16.4k, pick 15k
► Make time constant of output filter >10x oscillating freq
  • RL=4.7k, C2=10uF
► Set C1=C2

\[
F = \frac{1}{2*R4*C3*\ln(Vth/Vtl)}
\]

\[
F_c = \frac{1}{(2*\pi*RL*C2)}
\]
Comparator Example

Voltage Doubler - Results

► Trace 1 is 3.3V square wave (from R4 and C3)
► Trace 2 is charge on C1 (2*VCC – diode drop)
► Trace 3 is filtered output of doubler (2*VCC – 2 diode drops)
Emulated ADC

► Before using both pins are I/O
► ACMP+ is initially output low to discharge capacitor
► Comparator enabled when conversion needed
► ADC function compares ADC input voltage to voltage on C charged by R
► Timer is used to monitor time for RC to charge to ADC In
► Since RC is not linear a lookup table is used to adjust readings

Ref AN3266 – Getting Started with RS08
Emulated ADC Example using Analog Comparator

Implementation

► Define sampling time and timer resolution
► Define RC time constant
► Construct lookup table
► Define bus frequency
► Calibration

\[
\text{TimerResolution} = \frac{\text{ChargeUpTime}}{2^n - 1}
\]

ChargeUpTime chosen to be 1ms
n=8 for 8-bit timer (MTIM)
TimerResolution is 3.9us (round to 4us)
Max timer overflow = 255*4us= 1.02ms
Emulated ADC Example using Analog Comparator

Implementation

► Define sampling time and timer resolution
► Define RC time constant
► Construct lookup table
► Define bus frequency
► Calibration

\[
V = V_{DD} \left(1 - e^{-\frac{t}{RC}}\right)
\]

\[t = 4.61 \times RC\] for 99% charge
\[t = 1.02\text{ms} \Rightarrow\]

\[RC = \frac{\text{TimerOverflowPeriod}}{4.61} = 2.21 \times 10^{-4}\]

Given 1mA sink current at VDD = 5V, use 4.7kΩ
C is calculated to be 47nF
Emulated ADC Example using Analog Comparator

Implementation

► Define sampling time and timer resolution
► Define RC time constant
► Construct lookup table
► Define bus frequency
► Calibration

Non-Linearity Compensation Lookup Table

<table>
<thead>
<tr>
<th>Time (us)</th>
<th>ADC In</th>
<th>Timer Count</th>
<th>Linear ADC Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0.09</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>0.18</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>0.26</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>16</td>
<td>0.35</td>
<td>4</td>
<td>18</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1016</td>
<td>4.95</td>
<td>254</td>
<td>253</td>
</tr>
<tr>
<td>1020</td>
<td>4.95</td>
<td>255</td>
<td>253</td>
</tr>
</tbody>
</table>
Emulated ADC Example using Analog Comparator

Implementation

► Define sampling time and timer resolution
► Define RC time constant
► Construct lookup table
► Define bus frequency
► Calibration

• Choose at least 5x timer clock freq
• We chose 2MHz bus freq and set timer prescaler to divide-by-8 for a 250kHz timer clock.
• 250kHz gives 4us resolution
Emulated ADC Example using Analog Comparator

Implementation

► Define sampling time and timer resolution
► Define RC time constant
► Construct lookup table
► Define bus frequency
► Calibration
Crystal and PLL
Clock Modules on HC08, HCS08

- **Clock Generator Module (with PLL)**
  - LF – 32kHz
  - HF – 1 to 8MHz

- **Internal Clock Generator**
  - DCO and multiplier; External sources
  - FLL

- **Oscillator**
  - Internal sources
  - External divide by 4
  - Low power divide by 2

- **Internal Reference Clock (also Internal Clock Source)**
  - Internal reference
  - External sources
  - FLL
Crystal Components

1. What resistors should be used?

<table>
<thead>
<tr>
<th>Module</th>
<th>Y1</th>
<th>RS</th>
<th>Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
<td>typ</td>
<td>max</td>
</tr>
<tr>
<td>LF CGM</td>
<td>32.768kHz</td>
<td>220kΩ</td>
<td>330kΩ</td>
</tr>
<tr>
<td>HF CGM</td>
<td>1-8MHz</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ICG</td>
<td>1-10MHz</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>OSC, IRC</td>
<td>1MHz</td>
<td>20kΩ</td>
<td></td>
</tr>
<tr>
<td>OSC, IRC</td>
<td>4MHz</td>
<td>10kΩ</td>
<td></td>
</tr>
<tr>
<td>OSC, IRC</td>
<td>8-32MHz</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

- **RS** is a Series Resistor
- **RF** is a Feedback Resistor
- **RS** is used in low frequency Pierce oscillator configurations

<table>
<thead>
<tr>
<th>Module</th>
<th>Y1</th>
<th>RF</th>
<th>Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
<td>typ</td>
<td>max</td>
</tr>
<tr>
<td>LF CGM</td>
<td>32.768kHz</td>
<td>1MΩ</td>
<td>10MΩ</td>
</tr>
<tr>
<td>HF CGM</td>
<td>1 - 8MHz</td>
<td>0.5MΩ</td>
<td>1MΩ</td>
</tr>
<tr>
<td>ICG</td>
<td>1-10MHz</td>
<td>0.5MΩ</td>
<td>1MΩ</td>
</tr>
<tr>
<td>OSC, IRC</td>
<td>1-32MHz</td>
<td>0.5MΩ</td>
<td>1MΩ</td>
</tr>
</tbody>
</table>
2. What crystal should be used?

CL1 and CL2 are determined by the following formula:

\[ CL = \frac{CL1 \times CL2}{CL1 + CL2} + CS \]

CL is the Load Capacitance of the crystal

CS is stray capacitance

CL1 and CL2 are Load Capacitors

OSC1/EXTAL

OSC2/XTAL

MCU
Load Capacitance

CPIx is OSCx internal parasitic (6 to 9pF)

CPEx is OSCx external parasitic (0.5 to 1pF)

CPI1-2 is internal parasitic (0.25 to 0.5pF)

CPE1-2 is external parasitic (0.25 to 0.5pF)

\[ CL = \frac{C1 \times C2}{C1 + C2} + CS \]

where

\[ C1 = CL1 + CPI1 + CPE1 \]

\[ C2 = CL2 + CPI2 + CPE2 \]

\[ CS = CPI1-2 + CPE1-2 = 1pF \]

Internal parasitics are higher for physically larger packages
Load Capacitance – Pierce Example 1

Y1 = 32.768kHz, CL = 12.5pF

For best results pick CL1 ≤ CL2, within 1 or 2 std values

<table>
<thead>
<tr>
<th>Internal Parasitic Capacitance</th>
</tr>
</thead>
<tbody>
<tr>
<td>9pF</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>CL1</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>18</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Series</th>
<th>Frequency</th>
<th>Load Cap. (pF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFS145</td>
<td>32.768000 kHz</td>
<td>12.5</td>
</tr>
<tr>
<td>CFS206</td>
<td>32.768000 kHz</td>
<td>6.0</td>
</tr>
<tr>
<td>CFS308</td>
<td>32.768000 kHz</td>
<td>12.5</td>
</tr>
</tbody>
</table>

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2007-2008.
**Load Capacitance – Pierce Example 2**

**Y1 = 8.00MHz, CL = 20pF**

For best results pick CL1 ≤ CL2, within 1 or 2 std values.
All MCU crystals and ceramic resonators must be fundamental mode. HF ceramic resonators may also have spurs that can cause the oscillator to lock onto the wrong frequency.
CJM PLL Acquisition and Tracking

- Acquisition – CGMXFC filter makes large frequency corrections to VCO
- Tracking – CGMXFC filter makes small frequency corrections to VCO
- Auto bandwidth control – Lock detector switches between acquisition and tracking
- 3 component filter improves acquisition time and responds faster to disturbances
- 1 component filter is less expensive
PLL Filters

**HC08 LF**

**HC08 HF (A Family)**

**HC08 HF (Others)**

\[
C_F = C_{\text{fact}} \left( \frac{V_{\text{DDA}}}{f_{\text{CGMRD}}W} \right)
\]

**Table 4-5. Example Filter Component Values**

<table>
<thead>
<tr>
<th>(f_{\text{CLK}})</th>
<th>(C_{F1})</th>
<th>(C_{F2})</th>
<th>(R_{F1})</th>
<th>(C_F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 MHz</td>
<td>8.2 nF</td>
<td>820 pF</td>
<td>2k</td>
<td>18 nF</td>
</tr>
<tr>
<td>2 MHz</td>
<td>4.7 nF</td>
<td>470 pF</td>
<td>2k</td>
<td>6.8 nF</td>
</tr>
<tr>
<td>3 MHz</td>
<td>3.3 nF</td>
<td>330 pF</td>
<td>2k</td>
<td>5.6 nF</td>
</tr>
<tr>
<td>4 MHz</td>
<td>2.2 nF</td>
<td>220 pF</td>
<td>2k</td>
<td>4.7 nF</td>
</tr>
<tr>
<td>5 MHz</td>
<td>1.8 nF</td>
<td>180 pF</td>
<td>2k</td>
<td>3.9 nF</td>
</tr>
<tr>
<td>6 MHz</td>
<td>1.5 nF</td>
<td>150 pF</td>
<td>2k</td>
<td>3.3 nF</td>
</tr>
<tr>
<td>7 MHz</td>
<td>1.2 nF</td>
<td>120 pF</td>
<td>2k</td>
<td>2.7 nF</td>
</tr>
<tr>
<td>8 MHz</td>
<td>1 nF</td>
<td>100 pF</td>
<td>2k</td>
<td>2.2 nF</td>
</tr>
</tbody>
</table>

Figure 4-9. PLL Filter
Crystal and PLL Filter Layout Example

► 100nF bypass (C3) close to MCU
► No vias in crystal circuit
► VSS guard ring for crystal (no gnd loop)
► No high speed signals under crystal
► No ground plane under crystal
► Load capacitors (C1, C2) on crystal ground
► SMT jumpers (R0) to route VSS without vias
► Feedback (R1) and Series (R2) resistors in close proximity to crystal
Current Injection
What is Current Injection?

- Current injection is the current that is sourced into an I/O pin when the input voltage exceeds the power rails.
- Injected current can be positive or negative.
- Small amounts of injected current can be tolerated on digital I/O pins without affecting functionality.
- Injected current into an analog input pin can affect adjacent channel measurements.
- Current injection is a stress on the semiconductor.
# Current Injection Specs

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>Min</th>
<th>Typical</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC Injection Current A, B, C, D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single pin limit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{IN} &gt; V_{DD}$</td>
<td></td>
<td>0</td>
<td>-</td>
<td>2</td>
<td>mA</td>
</tr>
<tr>
<td>$V_{IN} &lt; V_{SS}$</td>
<td></td>
<td>0</td>
<td>-</td>
<td>-0.2</td>
<td>mA</td>
</tr>
<tr>
<td>Total MCU limit, includes sum of all stressed pins</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{IN} &gt; V_{DD}$</td>
<td></td>
<td>0</td>
<td>-</td>
<td>25</td>
<td>mA</td>
</tr>
<tr>
<td>$V_{IN} &lt; V_{SS}$</td>
<td></td>
<td>0</td>
<td>-</td>
<td>-5</td>
<td>mA</td>
</tr>
</tbody>
</table>

A This parameter is characterized and not tested on each device.

B All functional non-supply pins are internally clamped to $V_{SS}$ and $V_{DD}$.

C Input must be current limited to the value specified. To determine the value of the required current-limiting resistor, calculate resistance values for positive and negative clamp voltages, then use the larger of the two values.

D Power supply must maintain regulation within operating $V_{DD}$ range during instantaneous and operating maximum current conditions. If positive injection current ($V_{IN} > V_{DD}$) is greater than $I_{DD}$, the injection current may flow out of $V_{DD}$ and could result in external power supply going out of regulation. Ensure external $V_{DD}$ load will shunt current greater than maximum injection current. This will be the greatest risk when the MCU is not consuming power. Examples are: if no system clock is present, or if clock rate is very low (which would reduce overall power consumption).
Simplified I/O Pin Schematic

Primary paths for current injection

Secondary paths for current injection
Current Injection Example

► 70Vp-p AC signal from transformer
► I/O pin for zero crossing detect
► VDD supply = 5V
► Pick single series resistor of $R \geq 175k\ \Omega$
► **Tip** – pick order of magnitude higher $R$
► Best to have HV isolation (transformer, ferrites) or limiting components (diodes, EMI filters)

\[
\begin{align*}
+R &\geq (35V - 5V) / 2mA \\
+R &\geq 15k\Omega \\
-R &\geq (35V - 0V) / 0.2mA \\
-R &\geq 175k\Omega
\end{align*}
\]
Unused Pins
Unused Pin Termination

- Unused pins should be terminated to prevent current due to floating inputs

- Unused I/O pins
  - Leave unconnected, configure as outputs driving low
  - Leave unconnected, configure as inputs with pullups enabled
  - Use external pullups, configure as inputs

- Unused input pins
  - IRQ pin (dedicated or multiplexed) (Also RESET)
    - Use 4.7k to 10k pullup with 100nF to VSS
  - Use external pullups
  - Use internal pullups, if available

- Never connect unused pins directly to VDD or VSS
Unbonded Pin Termination

► Many MCUs have multiple packages and pin-counts
► All use the same die, so unbonded pins should also be terminated to prevent current due to floating inputs
► Configure as outputs driving high or low
► Configure as inputs with pull-ups enabled
Phototimer Example

► Circuit uses 8-pin 908QT2
► Port B pins are not bonded on the 8-pin package
► Terminate port B in firmware by
  • Configuring the port as outputs driving low or high
  • Configuring the port as inputs with the internal pullups enabled
► PTA3 is not used. Terminate PTA3 (mux’d RESET) by
  • Configuring as output driving high or low
  • Configuring as an input with the internal pull-up enabled

► For EMC/safety
  • Add 100nF cap to PTA2/IRQ
  • Add 10k, 100nF filter to PTA3/RST