EVALUATION KIT AVAILABLE

MAX5215/MAX5217

14-/16-Bit, Low-Power, Buffered Output, Rail-to-Rail DACs with I²C Interface

General Description

The MAX5215/MAX5217 are pin-compatible 14-bit and 16-bit digital-to-analog converters (DACs). The MAX5215/MAX5217 are single-channel, low-powered, buffered voltage-output DACs. The devices use a precision external reference applied through the high resistance input for rail-to-rail operation and low system power consumption. The MAX5215/MAX5217 accept a wide 2.7V to 5.5V supply voltage range. Power consumption is extremely low to accommodate most low-power and low-voltage applications.

The MAX5215/MAX5217 have an I²C-compatible, 2-wire serial interface that operates at clock rates up to 400kHz. On power-up, the MAX5215/MAX5217 reset the DAC output to zero, providing additional safety for applications that drive valves or other transducers that need to be off on power-up. The DAC output is buffered resulting in a low supply current of 80µA (max) and a low offset error of ±0.25mV. An asynchronous active-low input, $\overline{AUX}$, is provided. This input can be programmed to support clear or load DAC operations, independent of the serial interface. The MAX5215/MAX5217 are available in an ultra-small (3mm x 5mm), 8-pin µMAX® package and are specified over the -40°C to +105°C extended industrial temperature range.

Applications

Remote Sensing
Portable Instrumentation
Communication Systems
Automatic Tuning
Gain and Offset Adjustment
Power Amplifier Control
Automatic Test Equipment
Process Control and Servo Loops
Data Acquisition
Programmable Voltage and Current Sources

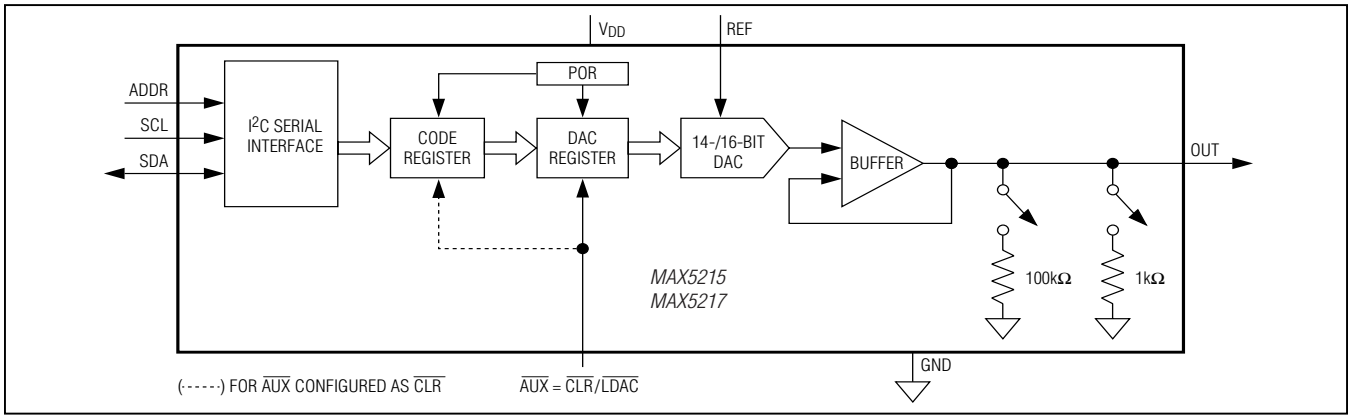
Features

- **Low-Power Consumption (80µA, max)**
- **18µs Settling Time**
- **16-/14-Bit Resolution in a 3mm x 5mm, 8-Pin µMAX Package**
- **Relative Accuracy**
  - ✧ **±0.4 LSB INL (MAX5215, 14 Bit) typ, 1 LSB (max)**
  - ✧ **±1.2 LSB INL (MAX5217, 16 Bit) typ, 4 LSB (max)**
- **Guaranteed Monotonic Over All Operating Range**
- **Low Gain and Offset Error**
- **Wide 2.7V to 5.5V Supply Range**
- **Rail-to-Rail Buffered Output Operation**
- **Safe Power-Up-Reset to Zero DAC Output**
- **I²C-Compatible 400kHz Serial Interface**
- **User-Programmable $\overline{AUX}$ Input Functions**
  - ✧ **$\overline{CLR}$, Clear to 0, Midscale, or Full Scale**
  - ✧ **$\overline{LDAC}$, Asynchronous Load DAC**
- **256kΩ Reference Input Resistance for Low-Power Operation**
- **Buffered Voltage Output Directly Drives 10kΩ Loads**
- **Output Power-Down Terminated with 1kΩ or 100kΩ to Ground or Left High Impedance**

***Ordering Information* appears at end of data sheet.**

µMAX is a registered trademark of Maxim Integrated Products, Inc.

Functional Block Diagram



For related parts and recommended products to use with this part, refer to: **www.maximintegrated.com/MAX5215.related**

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim's website at www.maximintegrated.com.

19-6469; Rev 0; 11/12

www.datasheetall.com

ABSOLUTE MAXIMUM RATINGS

V_{DD} to GND.....-0.3V to +6V
ADDR, REF, OUT,
$\overline{AUX}$ to GND .....-0.3V to the lower of (V_{DD} + 0.3V) and +6V
SCL, SDA, to GND .....-0.3V to +6V
Continuous Power Dissipation (T_A = +70°C)
µMAX (derate at 4.8mW/°C above 70°C).....387mW
Maximum Current into Any Input or Output..... ±50mA
Operating Temperature Range..... -40°C to +105°C
Storage Temperature Range..... -65°C to +150°C
Lead Temperature (soldering, 10s) .....+300°C
Soldering Temperature (reflow) .....+260°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

PACKAGE THERMAL CHARACTERISTICS (Note 1)

µMAX
Junction-to-Ambient Thermal Resistance (θ_{JA}) .....206°C/W
Junction-to-Case Thermal Resistance (θ_{JC}) .....42°C/W

Note 1: Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to **www.maximintegrated.com/thermal-tutorial**.

ELECTRICAL CHARACTERISTICS

(V_{DD} = 2.7V to 5.5V, V_{REF} = 2.5V to V_{DD}, C_L = 60pF, R_L = 10kΩ, T_A = -40°C to 105°C, unless otherwise noted. Typical values are at T_A = +25°C.)(Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
STATIC ACCURACY (Note 3)						
Resolution	N	MAX5215	14			Bits
		MAX5217/MAX5217B	16			
Integral Nonlinearity	INL	MAX5215 (14 bit) (Note 4)	-1	±0.4	+1	LSB
		MAX5217 (16 bit) (Note 4)	-4	±1.2	+4	
		MAX5217B (16 bit) (Note 4)	-8	± 3	+8	
Differential Nonlinearity	DNL	MAX5215 (14 bit) (Note 4)	-1	±0.1	+1	LSB
		MAX5217/5217B (16 bit) (Note 4)	-1	±0.25	+1	
Offset Error	OE	MAX5215/5217 (Note 5)	-1.25	±0.25	+1.25	mV
		MAX5217B (Note 5)	-3	±0.5	-3	
Offset-Error Drift				±1.6		µV/°C
Gain Error	GE	MAX5215/5217 (Note 5)	-0.06	-0.04	0	%FS
		MAX5217B (Note 5)	-0.10	-0.04	0	
Gain Temperature Coefficient				±2		ppm FS/°C
REFERENCE INPUT						
Reference-Input Voltage Range	V_{REF}		2		V_{DD}	V
Reference-Input Impedance	R_{REF}		200	256		kΩ
DAC OUTPUT						
Output Voltage Range (Note 6)		No load	0		V_{DD}	V
		10kΩ load to GND	0		V_{DD} - 0.2	
		10kΩ load to V_{DD}	0.2		V_{DD}	
DC Output Impedance				0.1		Ω

ELECTRICAL CHARACTERISTICS (continued)

(V_{DD} = 2.7V to 5.5V, V_{REF} = 2.5V to V_{DD}, C_L = 60pF, R_L = 10kΩ, T_A = -40°C to 105°C, unless otherwise noted. Typical values are at T_A = +25°C.)(Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Maximum Capacitive Load (No Sustained Oscillations)	C_L	Series resistance = 0Ω		0.1		nF
		Series resistance = 1kΩ		15		µF
Resistive Load (Note 7)	R_L		5			kΩ
Short-Circuit Current		V_{DD} = 5.5V	-25	±6	+25	mA
Power-Up Time		From power-down mode		25		µs
DYNAMIC PERFORMANCE (Note 7)						
Voltage-Output Slew Rate	SR	Positive and negative		0.5		V/µs
Voltage-Output Settling Time		¼ scale to ¾ scale, to ±0.5 LSB, 14 bit.		18		µs
Reference –3dB Bandwidth	BW	Hex code = 2000 (MAX5215), Hex code = 8000 (MAX5217)		100		kHz
Digital Feedthrough		Code = 0, all digital inputs from 0V to V_{DD}, SCL < 400kHz		1.0		nV·s
DAC Glitch Impulse		Major code transition		5		nV·s
Output Noise		1kHz		73		nV/√Hz
		10kHz		70		
Integrated Output Noise		0.1Hz to 10Hz		3.5		μV_{P-P}
POWER REQUIREMENTS						
Supply Voltage	V_{DD}		2.7		5.5	V
Supply Current	I_{DD}	No load; all digital inputs at 0V or V_{DD}, supply current only; excludes reference input current.		70	80	µA
Power-Down Supply Current	PDI_{DD}	No load, all digital inputs at 0V or V_{DD}		0.4	2	µA
DIGITAL INPUTS (SCL, SDA, $\overline{AUX}$, ADDR)						
Input High Voltage	V_{IH}		0.7 x V_{DD}			V
Input Low Voltage	V_{IL}				0.3 x V_{DD}	V
Hysteresis Voltage	V_{HYS}			0.15		V
Input Leakage Current	I_{IN}	V_{IN} = 0V or V_{DD}		±0.1	±1	µA
Input Capacitance (Note 7)	C_{IN}				10	pF
ADDR Pullup/Pulldown Strength		(Note 8)	30	50	90	kΩ
DIGITAL OUTPUT (SDA)						
Output Low Voltage	V_{OL}	I_{SINK} = 3mA			0.2	V

ELECTRICAL CHARACTERISTICS (continued)

(V_{DD} = 2.7V to 5.5V, V_{REF} = 2.5V to V_{DD}, C_L = 60pF, R_L = 10kΩ, T_A = -40°C to 105°C, unless otherwise noted. Typical values are at T_A = +25°C.)(Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
TIMING CHARACTERISTICS						
SCL Clock Frequency	f_{SCL}				400	kHz
sBus Free Time Between a STOP and a START Condition	t_{BUF}		1.3			µs
Hold Time for a Repeated START Condition	$t_{HD;STA}$		0.6			µs
SCL Pulse Width Low	t_{LOW}		1.3			µs
SCL Pulse Width High	t_{HIGH}		0.6			µs
Setup Time for Repeated START Condition	$t_{SU;STA}$		0.6			µs
Data Hold Time	$t_{HD;DAT}$		0		900	ns
Data Setup Time	$t_{SU;DAT}$		100			ns
SDA and SCL Receiving Rise Time	t_r		20 + C_B/10		300	ns
SDA and SCL Receiving Fall Time	t_f		20 + C_B/10		300	ns
SDA Transmitting Fall Time	t_f		20 + C_B/10		250	ns
Setup Time for STOP Condition	$t_{SU;STO}$		0.6			µs
Bus Capacitance Allowed	C_B	V_{DD} = 2.7V to 5.5V	10		400	pF
Pulse Width of Suppressed Spike	t_{SP}			50		ns
$\overline{CLR}$ Removal Time Prior to a Recognized START	t_{CLRSTA}		100			ns
$\overline{CLR}$ Pulse Width Low	t_{CLPW}		20			ns
$\overline{LDAC}$ Pulse Width Low	t_{LDPW}		20			ns
SCLK Rise to $\overline{LDAC}$ Fall Hold	t_{LDH}	Applies to execution edge	400			ns

Note 2: Electrical specifications are production tested at T_A = +25°C and T_A = +105°C. Specifications over the entire operating temperature range are guaranteed by design and characterization. Typical specifications are at T_A = +25°C and are not guaranteed.

Note 3: Static accuracy tested without load.

Note 4: Linearity is tested within 20mV of GND and V_{DD}.

Note 5: Gain and offset is tested within 20mV of GND and V_{DD}.

Note 6: Subject to offset and gain error limits and V_{REF} settings.

Note 7: Specification is guaranteed by design and characterization.

Note 8: Unconnected conditions on the ADDR_ inputs are sensed through a resistive pullup and pulldown operation; for proper operation, the ADDR_ inputs must be connected to V_{DD}, GND, or left unconnected with minimal capacitance.

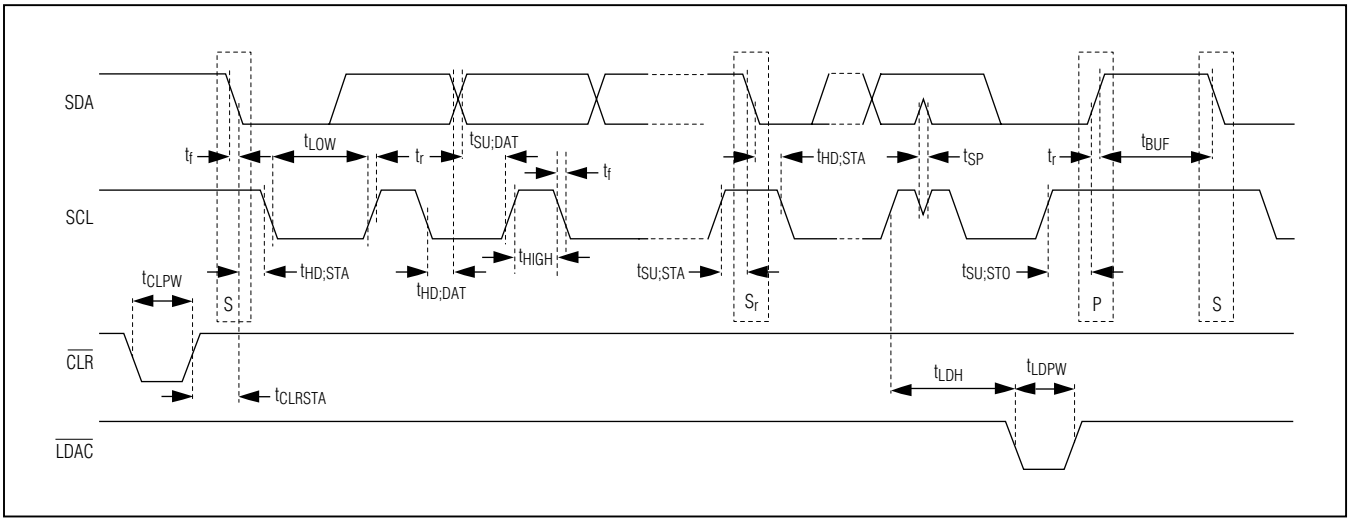
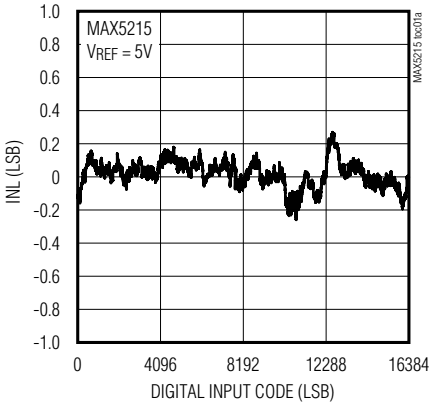


Figure 1. I²C Serial Interface Timing Diagram

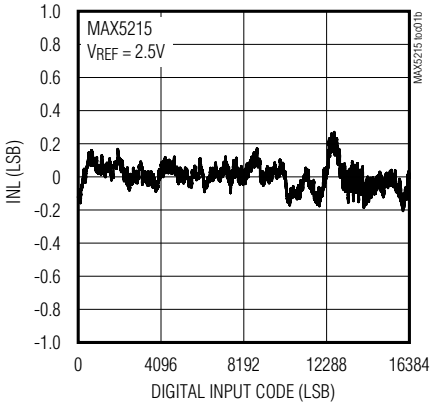
Typical Operating Characteristics

(V_{DD} = 5V, T_A = +25°C, unless otherwise noted.)

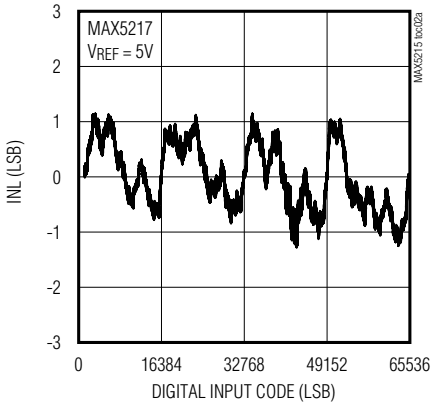
INTEGRAL NONLINEARITY vs. DIGITAL INPUT CODE



INTEGRAL NONLINEARITY vs. DIGITAL INPUT CODE



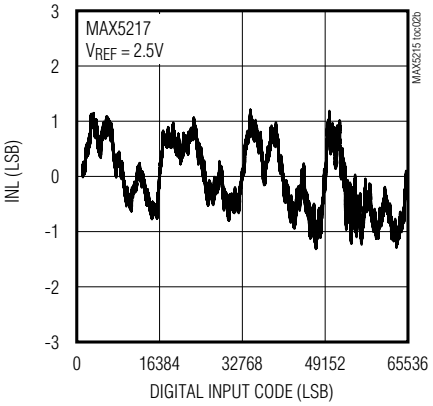
INTEGRAL NONLINEARITY vs. DIGITAL INPUT CODE



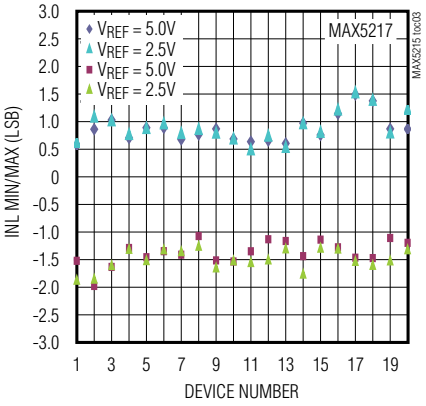
Typical Operating Characteristics (continued)

(V_{DD} = 5V, T_A = +25°C, unless otherwise noted.)

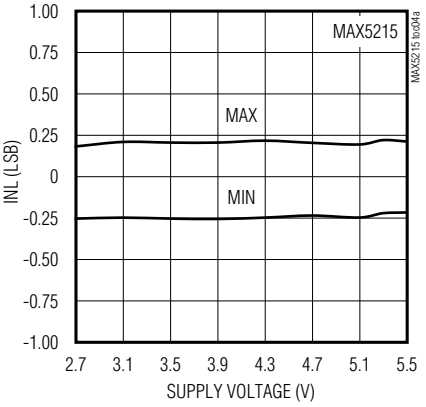
INTEGRAL NONLINEARITY vs. DIGITAL INPUT CODE



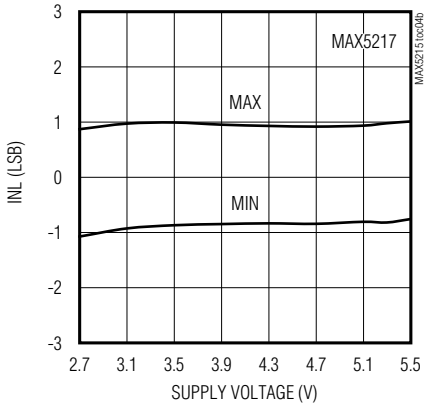
INL MIN/MAX (V_{REF} = 5.0V/2.5V)



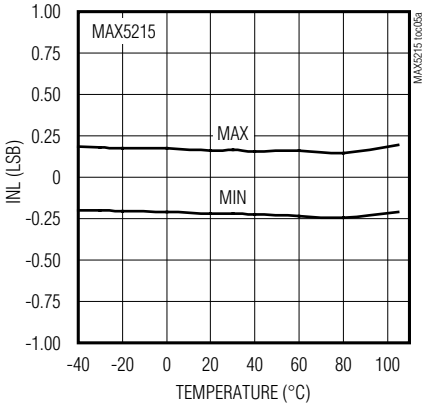
INTEGRAL NONLINEARITY vs. SUPPLY VOLTAGE



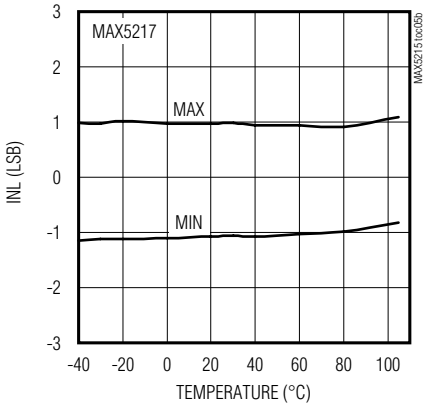
INTEGRAL NONLINEARITY vs. SUPPLY VOLTAGE



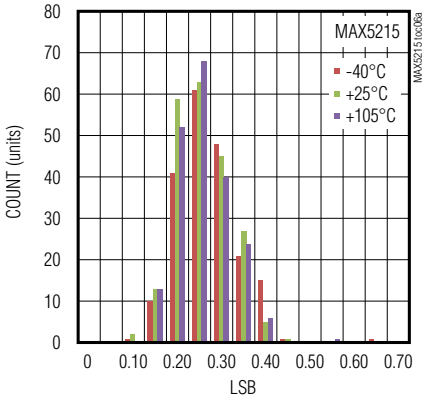
INTEGRAL NONLINEARITY vs. TEMPERATURE



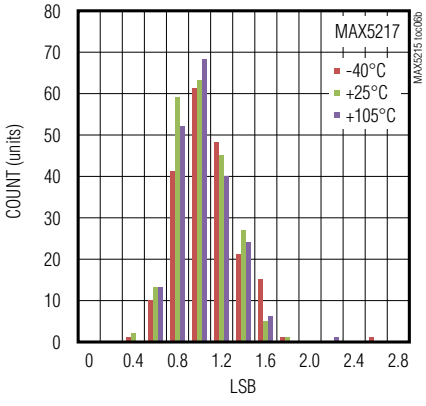
INTEGRAL NONLINEARITY vs. TEMPERATURE



MAX(ABS(INL)) DISTRIBUTION vs. TEMPERATURE



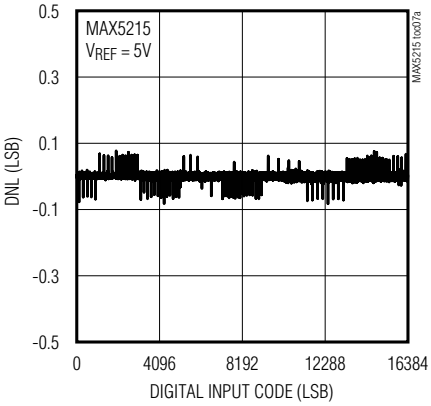
MAX(ABS(INL)) DISTRIBUTION vs. TEMPERATURE



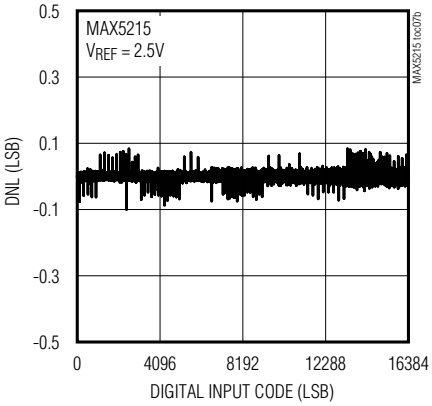
Typical Operating Characteristics (continued)

(V_{DD} = 5V, T_A = +25°C, unless otherwise noted.)

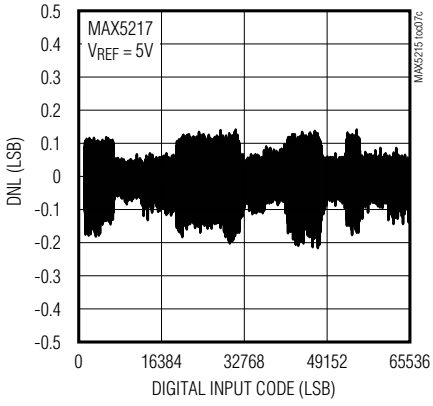
DIFFERENTIAL NONLINEARITY vs. DIGITAL INPUT CODE



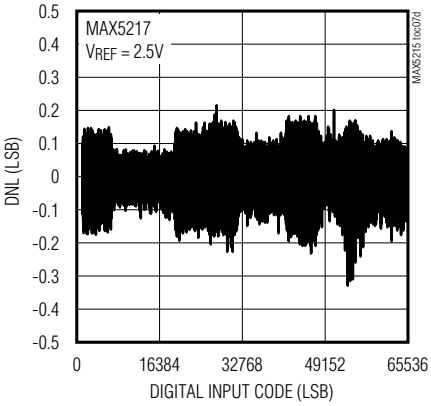
DIFFERENTIAL NONLINEARITY vs. DIGITAL INPUT CODE



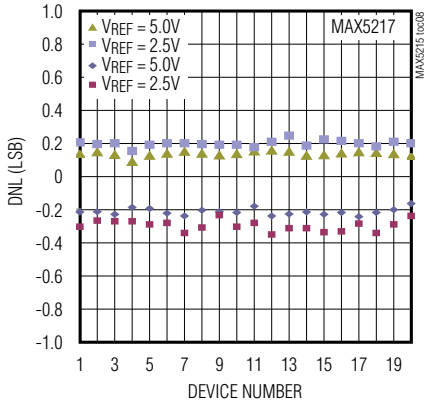
DIFFERENTIAL NONLINEARITY vs. DIGITAL INPUT CODE



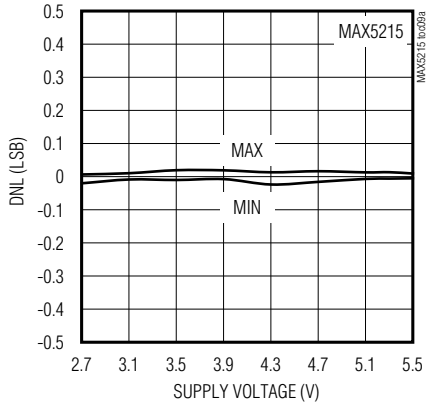
DIFFERENTIAL NONLINEARITY vs. DIGITAL INPUT CODE



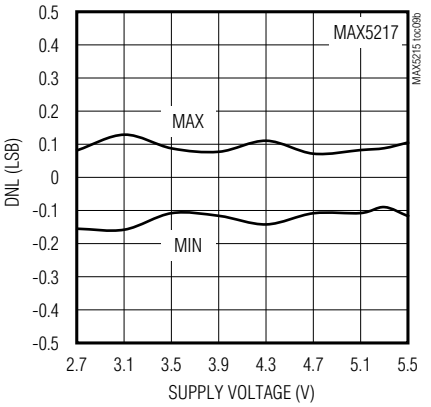
DNL MIN/MAX (V_{REF} = 5.0V/2.5V)



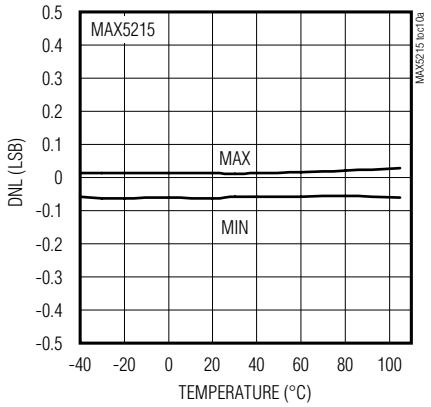
DIFFERENTIAL NONLINEARITY vs. SUPPLY VOLTAGE



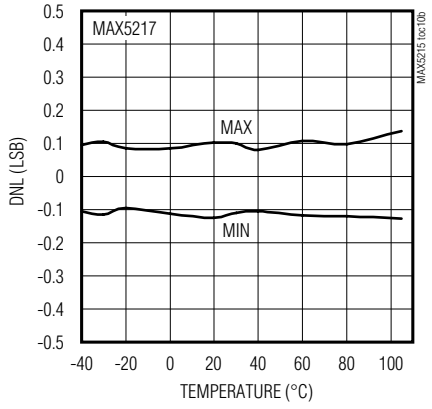
DIFFERENTIAL NONLINEARITY vs. SUPPLY VOLTAGE



DIFFERENTIAL NONLINEARITY vs. TEMPERATURE



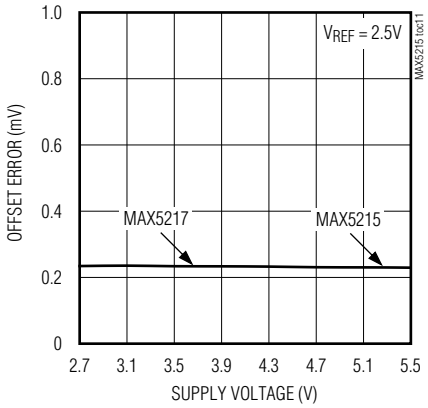
DIFFERENTIAL NONLINEARITY vs. TEMPERATURE



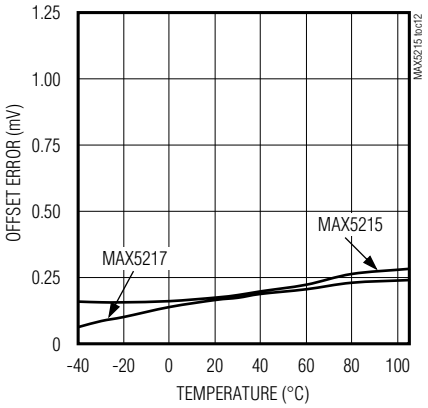
Typical Operating Characteristics (continued)

(V_{DD} = 5V, T_A = +25°C, unless otherwise noted.)

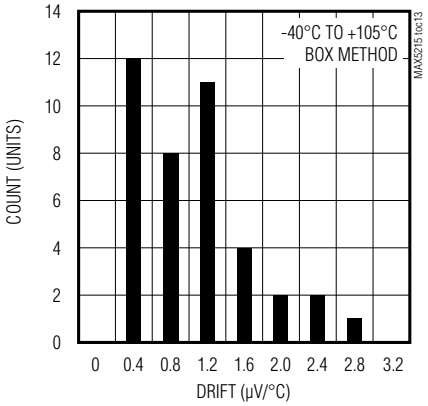
OFFSET ERROR vs. SUPPLY VOLTAGE



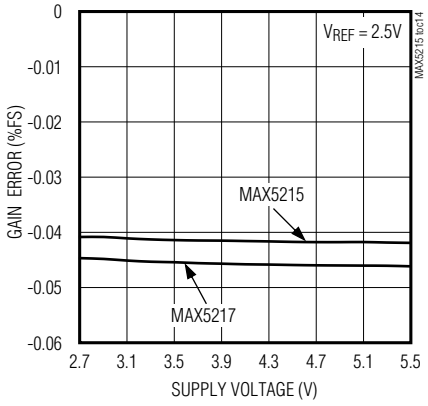
OFFSET ERROR vs. TEMPERATURE



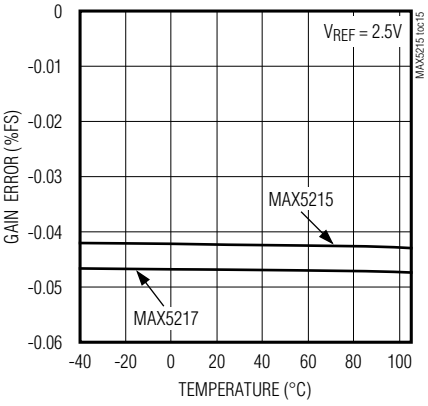
OFFSET ERROR DRIFT vs. TEMPERATURE DISTRIBUTION



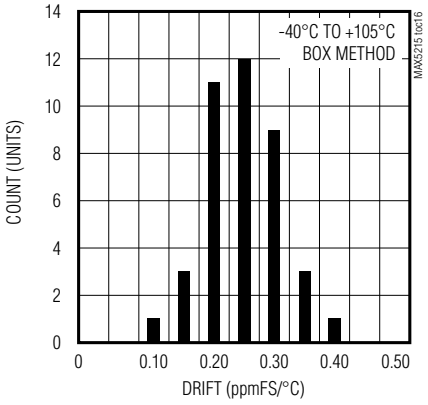
GAIN ERROR vs. SUPPLY



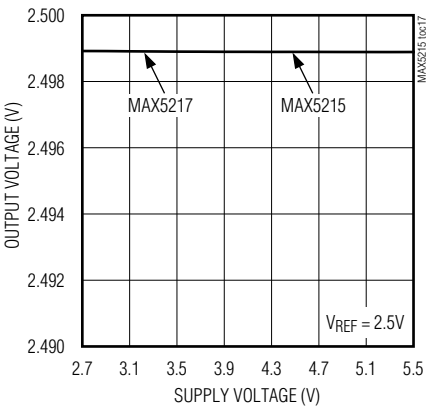
GAIN ERROR vs. TEMPERATURE



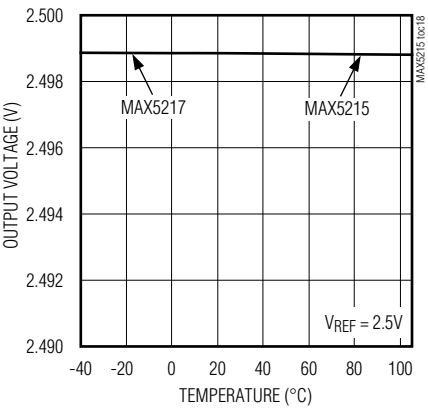
GAIN ERROR DRIFT vs. TEMPERATURE DISTRIBUTION



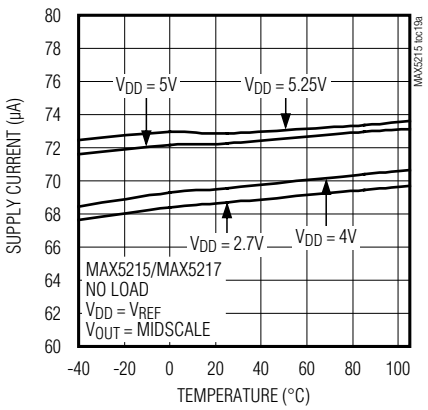
FULL-SCALE OUTPUT vs. SUPPLY VOLTAGE



FULL-SCALE OUTPUT vs. TEMPERATURE



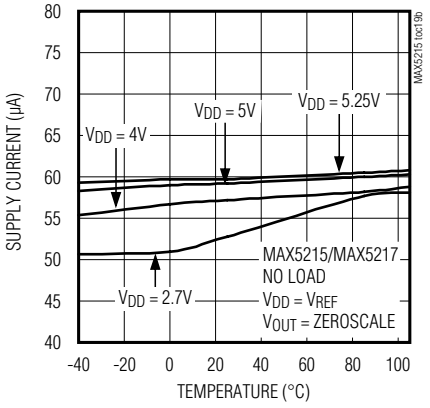
SUPPLY CURRENT vs. TEMPERATURE



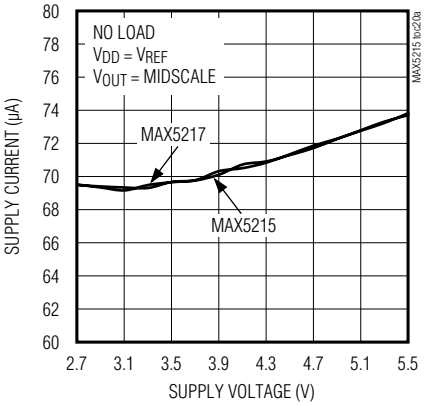
Typical Operating Characteristics (continued)

(V_{DD} = 5V, T_A = +25°C, unless otherwise noted.)

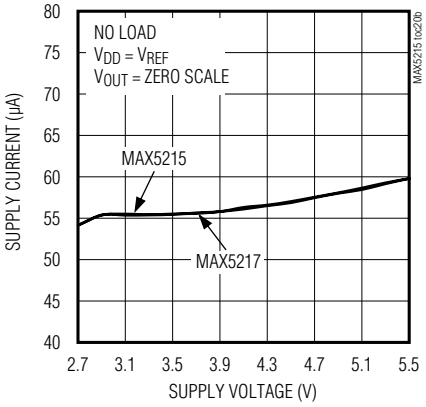
SUPPLY CURRENT vs. TEMPERATURE



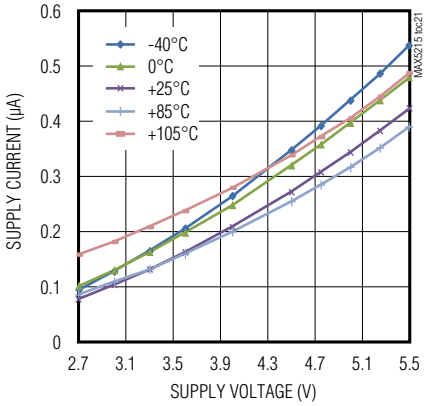
SUPPLY CURRENT vs. SUPPLY VOLTAGE



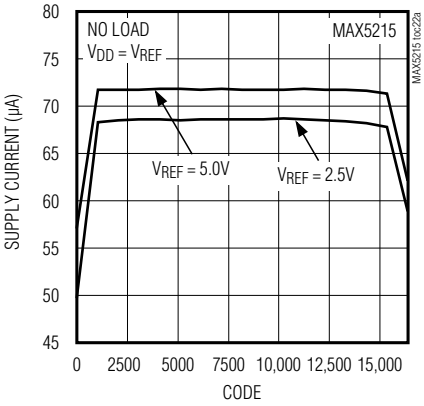
SUPPLY CURRENT vs. SUPPLY VOLTAGE



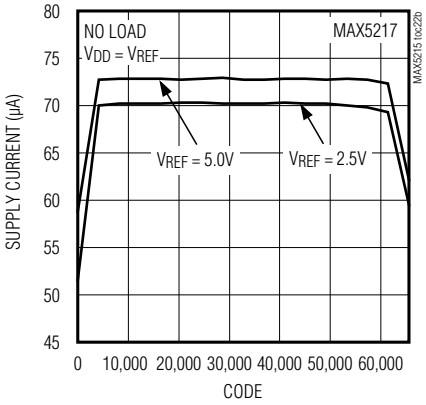
SUPPLY CURRENT vs. SUPPLY VOLTAGE (POWER-DOWN MODE)



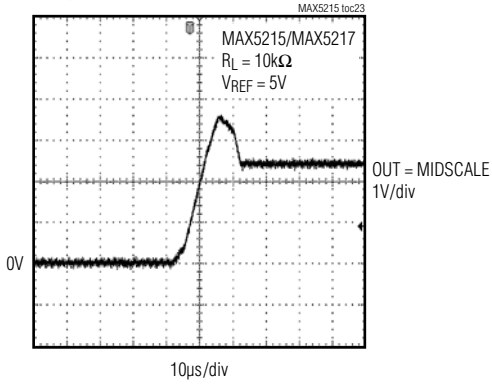
SUPPLY CURRENT vs. DAC CODE



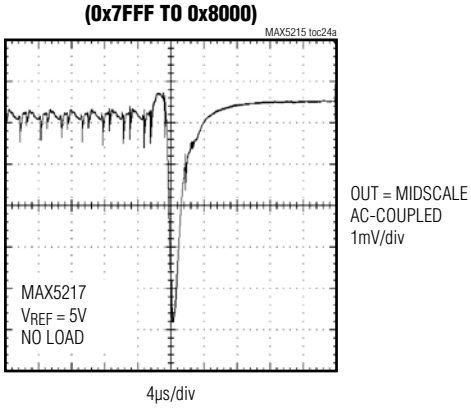
SUPPLY CURRENT vs. DAC CODE



V_{OUT} vs. TIME (EXITING POWER-DOWN MODE)



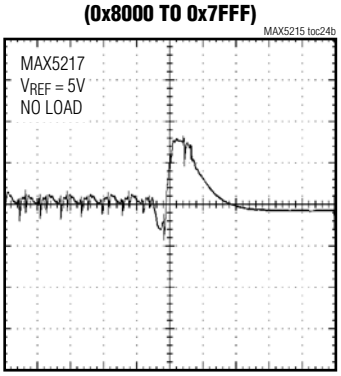
MAJOR CODE TRANSITION (0x7FFF TO 0x8000)



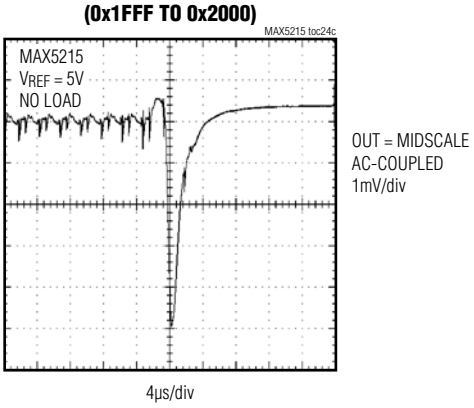
Typical Operating Characteristics (continued)

(V_{DD} = 5V, T_A = +25°C, unless otherwise noted.)

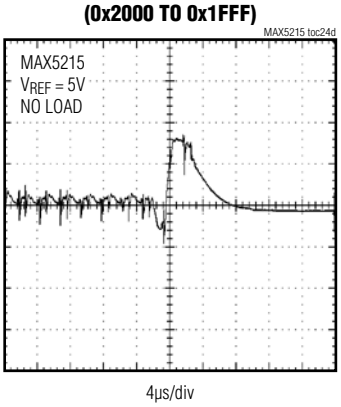
MAJOR CODE TRANSITION (0x8000 TO 0x7FFF)



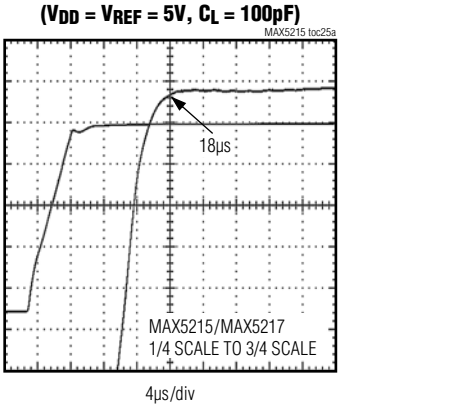
MAJOR CODE TRANSITION (0x1FFF TO 0x2000)



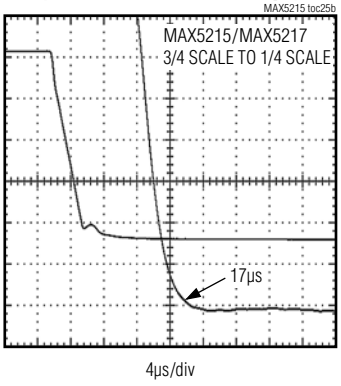
MAJOR CODE TRANSITION (0x2000 TO 0x1FFF)



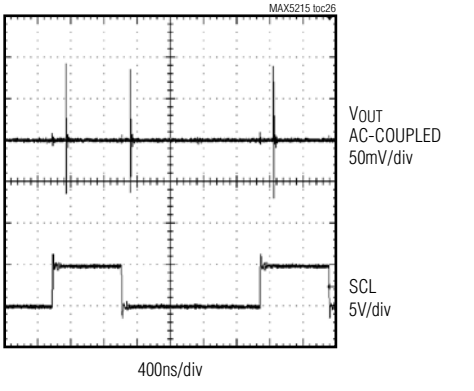
SETTLING TO ±0.5 LSB 14 BIT (V_{DD} = V_{REF} = 5V, C_L = 100pF)



SETTLING TO ±0.5 LSB 14 BIT (V_{DD} = V_{REF} = 5V, C_L = 100pF)



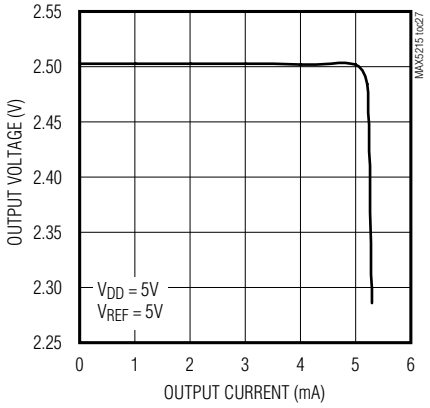
DIGITAL FEEDTHROUGH



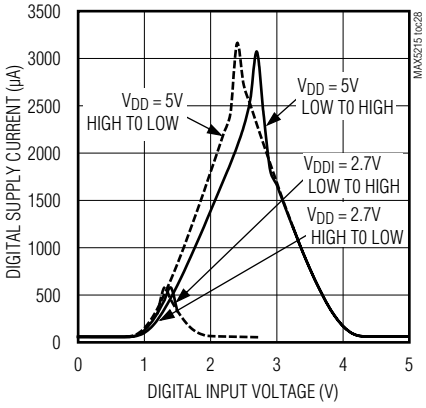
Typical Operating Characteristics (continued)

(V_{DD} = 5V, T_A = +25°C, unless otherwise noted.)

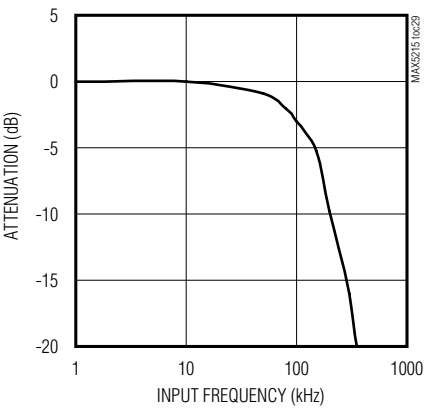
OUTPUT VOLTAGE vs. OUTPUT CURRENT



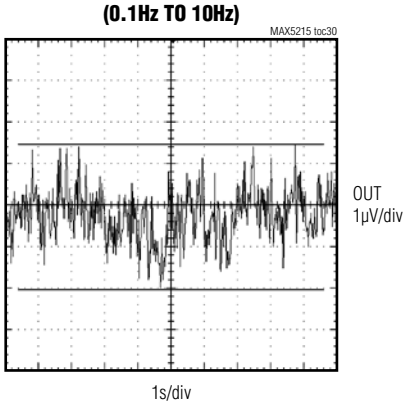
SUPPLY CURRENT vs. DIGITAL INPUT VOLTAGE



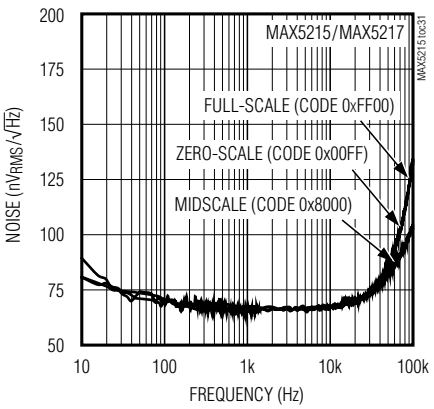
REFERENCE INPUT BANDWIDTH vs. FREQUENCY



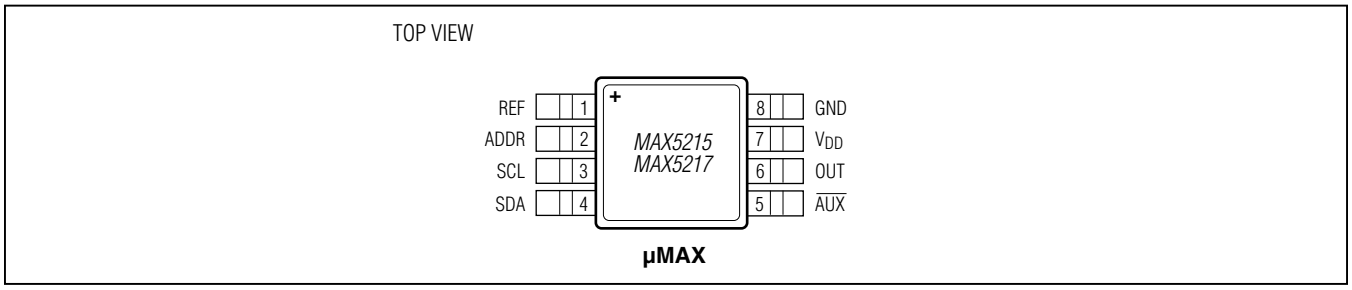
INTEGRATED OUTPUT NOISE (0.1Hz TO 10Hz)



DAC OUPUT NOISE DENSITY vs. FREQUENCY



Pin Configuration



Pin Description

PIN	NAME	FUNCTION
1	REF	Reference Voltage Input. Bypass REF with a 0.1µF capacitor to GND.
2	ADDR	I²C Device Address Input. Pull high, low, or do not connect to set the two LSBs of the device address.
3	SCL	I²C Serial Clock Input
4	SDA	I²C Serial Data Input
5	$\overline{AUX}$	User-Configurable Active-Low Asynchronous Input. When configured as $\overline{CLR}$ mode: drive $\overline{AUX}$ low to clear the contents of the input CODE and the DAC registers and return the DAC to a user-selectable return state (default). When configured as $\overline{LDAC}$ mode: drive $\overline{AUX}$ low to load the pending CODE register content to the active DAC register.
6	OUT	Buffered DAC Voltage Output
7	V_{DD}	Supply Voltage. Bypass V_{DD} with a 0.1µF capacitor to GND.
8	GND	Ground

Detailed Description

The MAX5215/MAX5217 are 14-bit and 16-bit single-channel, low-power, high reference input resistance, buffered voltage-output DACs. These devices feature a fast 400kHz I²C serial interface. The MAX5215/MAX5217 include a serial-in/parallel-out shift register, internal CODE and DAC registers, a power-on-reset (POR) circuit to initialize the DAC output to code zero, and an output buffer to allow rail-to-rail operation. The 2.7V to 5.5V wide supply voltage range and low-power consumption accommodate most low-power and low-voltage applications. On power-up, the MAX5215/MAX5217 reset the DAC output to zero, providing additional safety for applications that drive valves or other transducers that need to be off during power-up.

The MAX5215/MAX5217 feature a configurable asynchronous active-low input ($\overline{AUX}$) that can be programmed by the user to act as either an asynchronous clear input ($\overline{CLR}$) or a load DAC input ($\overline{LDAC}$). By default, the devices operate in $\overline{CLR}$ mode on power-up.

DAC Output (OUT)

The MAX5215/MAX5217 include an internal buffer on the DAC output. The internal buffer provides improved load regulation and transition glitch suppression for the DAC output. The output buffer slews at 0.5V/µs and drives up to 10kΩ in parallel with 100pF. The analog supply voltage (V_{DD}) determines the maximum output voltage range of the device as V_{DD} powers the output buffer. Under no-load conditions, the output buffer drives from GND to V_{DD}, subject to offset and gain errors. With a 10kΩ load to GND, the output buffer drives from GND to within 200mV of V_{DD}. With a 10kΩ load to V_{DD}, the output buffer drive from V_{DD} to within 200mV of GND.

The DAC ideal output voltage is defined by:

$$V_{OUT} = V_{REF} \times D/2^N$$

where D = code loaded into the DAC register, V_{REF} = reference voltage, N = resolution

DAC Reference (REF)

The external reference input features a typical input impedance of 256kΩ (independent of the DAC code) and accepts an input voltage from +2V to V_{DD}. Connect an external voltage supply between REF and GND to apply an external reference. Visit **www.maximintegrated.com/products/references** for a list of available voltage-reference devices.

Internal Register Structure

The user interface is separated from the DAC logic to minimize digital feedthrough. Within the serial interface is an input shift register, the contents of which can be routed to the control register or DAC, as determined by the user command.

Within the device, there is a CODE register followed by a DAC Latch register (see the *Functional Diagram*). The contents of the CODE register hold pending DAC output settings which can later be loaded into the DAC register. The CODE register can be updated using both CODE and CODE_LOAD user commands. The contents of the DAC register hold the current DAC output settings. The DAC register can be updated directly from the serial interface using the CODE_LOAD commands or can upload the current contents of the CODE register using LOAD commands or the $\overline{LDAC}$ logic input.

The contents of both CODE and DAC registers are maintained during power-down states, so that when the DAC is powered on, the previously stored output setting is restored. Any CODE or LOAD commands issued during power-down states continue to update the register contents.

$\overline{AUX}$ Configured as Clear Input, $\overline{CLR}$

When configured in $\overline{CLR}$ mode, the $\overline{AUX}$ input performs an asynchronous level sensitive CLEAR operation. If $\overline{CLR}$ is pulled low, the CODE and DAC data registers are reset to their clear values as defined by the user configuration settings (see Table 9). User configuration settings are not affected.

If $\overline{CLR}$ is asserted at any point during an I²C write sequence, from that point on, and until $\overline{CLR}$ is deasserted, all I²C commands attempting to modify CODE or DAC register contents are ignored. The $\overline{CLR}$ activity is given precedence and the commands are gated. In all cases, the I²C interface continues to function according to protocol, however slave ACK pulses beyond the command byte acknowledge will not be sent for gated command sequences (notifying the µP that the gated instructions are being ignored). This gating condition remains in effect until the $\overline{CLR}$ condition is removed and a subsequent I²C START condition is recognized (beginning a new I²C write sequence), meeting t_{CLRSTA} requirements (Figure 1). If $\overline{CLR}$ is driven low during an I²C command read sequence, the exchange continues as normal, however the data being read back may be stale, having since been cleared. The user may determine the state of the $\overline{CLR}$ pin by issuing a Part ID read command.

An equivalent software clear operation is provided through the SW_CLEAR command.

$\overline{AUX}$ Configured as Load DAC Input, $\overline{LDAC}$

When configured in $\overline{LDAC}$ mode, the $\overline{AUX}$ input performs an asynchronous level sensitive LOAD operation when it is pulled low. Internally, a dual register system is provided, with pending DAC output settings stored in a CODE register, while the current output settings are stored in the DAC latches. When $\overline{LDAC}$ is pulled low, the DAC latches are held in a transparent state, and the CODE register contents are loaded and stored. This allows several DACs to be updated simultaneously using a common $\overline{LDAC}$ line, or allows the DAC to be quickly updated to a pending setting via a single pin operation.

Users wishing to load new DAC data in direct response to I²C activity can enable and connect $\overline{LDAC}$ permanently low. Users wishing to control the DAC update instance independently of the I²C instruction should hold $\overline{LDAC}$ high during programming cycles. Once the programming is complete, $\overline{LDAC}$ may be strobed and the new DAC codes will be loaded (this method allows simultaneous updates of several devices). Be sure to observe the t_{LDH} timing requirements (Figure 1).

A software load operation is provided through the LOAD or CODE_LOAD command. With the software load operation, the content of the CODE register will be latched into the DAC register regardless of the status and configuration of the $\overline{LDAC}$ pin. Multiple MAX5215/MAX5217 can be loaded synchronously using software load commands in conjunction with the Broadcast ID.

I²C Serial Interface

The MAX5215/MAX5217 feature an I²C/SMBus™-compatible 2-wire serial interface consisting of a serial data line (SDA) and a serial clock line (SCL). SDA and SCL enable communication between the part and the master at clock rates up to 400kHz. Figure 1 shows the 2-wire interface timing diagram. The master generates SCL and initiates data transfer on the bus. The master device writes data to the MAX5215/MAX5217 by transmitting the proper slave address followed by the register address and then the data word. Each transmit sequence is framed by a START (S) or Repeated START (Sr) condition and a STOP (P) condition. Each word transmitted to the part is 8 bits long and is followed by an acknowledge clock pulse.

A master reading data from the MAX5215/MAX5217 must transmit the proper slave address followed by a series of nine SCL pulses for each byte of data requested. The MAX5215/MAX5217 transmit data on SDA in sync with the master-generated SCL pulses. The master acknowledges receipt of each byte of data. Each read sequence is framed by a START or Repeated START condition, a not acknowledge, and a STOP condition.

SDA operates as both an input and an open-drain output. A pullup resistor, typically 4.7kΩ, is required on SDA. SCL operates only as an input. A pullup resistor, typically 4.7kΩ, is required on SCL if there are multiple masters on the bus, or if the single master has an open-drain SCL output.

Series resistors in line with SDA and SCL are optional. Series resistors protect the digital inputs of the MAX5215/MAX5217 from high voltage spikes on the bus lines, and minimize crosstalk and undershoot of the bus signals. The MAX5215/MAX5217 can accommodate bus voltages higher than V_{DD} up to a limit of 5.5V; bus voltages lower than V_{DD} are not recommended and may result in significantly increased interface currents.

SMBus is a trademark of Intel Corp.

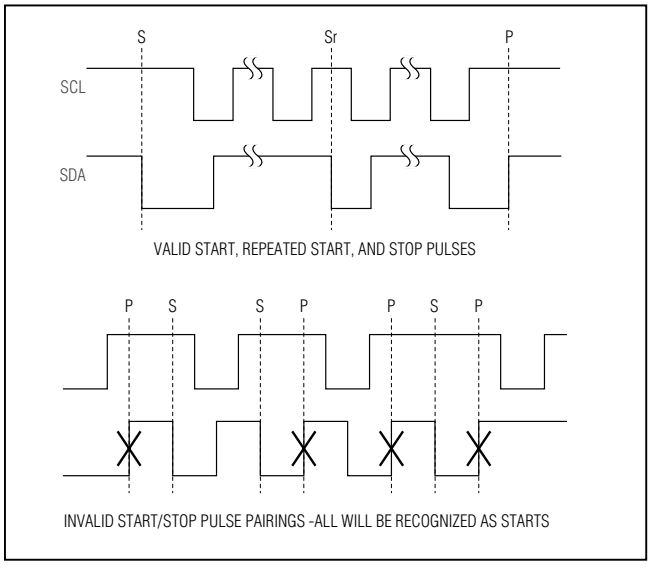


Figure 2. I²C START, Repeated START, and STOP Conditions

I²C Bit Transfer

One data bit is transferred during each SCL cycle. The data on SDA must remain stable during the high period of the SCL pulse. Changes in SDA while SCL is high are control signals (see the *I²C START and STOP Conditions* section).

I²C START and STOP Conditions

SDA and SCL idle high when the bus is not in use. A master initiates communication by issuing a START condition. A START condition is a high-to-low transition on SDA with SCL high. A STOP condition is a low-to-high transition on SDA while SCL is high (Figure 2). A START condition from the master signals the beginning of a transmission to the MAX5215/MAX5217. The master terminates transmission, and frees the bus, by issuing a STOP condition. The bus remains active if a Repeated START condition is generated instead of a STOP condition.

I²C Early STOP and Repeated START Conditions

The MAX5215/MAX5217 recognize a STOP condition at any point during data transmission except if the STOP condition occurs in the same high pulse as a START condition. For proper operation, do not send a STOP condition during the same SCL high pulse as the START condition. Transmissions ending in an early STOP condition will not impact the internal device settings. If the STOP occurs during a readback byte, the transmission is terminated and a later read mode request will begin transfer of the requested register data from the beginning.

Table 1. Two LSBs of the Slave Address Determined by the ADDR Input

ADDR	A1	A0
GND	0	0
N.C.	0	1
V_{DD}	1	1

I²C Slave Address

The slave address is defined as the seven most significant bits (MSBs) followed by the R/$\overline{W}$ bit (Figure 1). The 5 MSBs (A[6:2]) are 00111 with the two LSBs (A[1:0]) determined by the input ADDR as shown in Table 1. Setting the R/$\overline{W}$ bit to 1 configures the MAX5215/MAX5217 for read mode. Setting the R//W bit to 0 configures the MAX5215/MAX5217 for write mode. The slave address is the first byte of information sent to the MAX5215/MAX5217 after the START condition.

The MAX5215/MAX5217 have the ability to detect an unconnected state on the ADDR input for additional address flexibility; if leaving the ADDR input unconnected, be certain to minimize all loading on the pin (i.e. provide a landing for the pin, but do not any board traces.

I²C Broadcast Address

A broadcast address is provided for the purpose of updating or configuring all MAX5215/MAX5217 devices on a given I²C bus. All MAX5215/MAX5217 acknowledge and respond to the broadcast device address 01010100 regardless of the state of the address input pin. The broadcast is intended for use in write mode only (as indicated by R/$\overline{W}$ = 0 in the address given)

I²C Acknowledge

In write mode, the acknowledge bit (ACK) is a clocked 9th bit that the MAX5215/MAX5217 use to handshake receipt of each byte of data when in write mode as shown in Figure 3. The MAX5215/MAX5217 pull down SDA during the entire master-generated 9th clock pulse if the previous byte is successfully received. Monitoring ACK allows for detection of unsuccessful data transfers. An unsuccessful data transfer occurs if a receiving device is busy or if a system fault has occurred. In the event of an unsuccessful data transfer, the bus master will retry communication.

In read mode, the master pulls down SDA during the 9th clock cycle to acknowledge receipt of data when the MAX5215/MAX5217 are in read mode. An acknowledge is sent by the master after each read byte to allow data transfer to continue. A not-acknowledge is sent when the master reads the final byte of data from the MAX5215/MAX5217, followed by a STOP condition.

I²C Write Operation (Standard Protocol)

A master device communicates with the MAX5215/MAX5217 by transmitting the proper slave address followed by command and data words. Each transmit sequence is framed by a START or Repeated START condition and a STOP condition as described above. Each word is 8 bits long and is always followed by an acknowledge clock (ACK) pulse as shown in Figure 4and Figure 5. The first byte contains the address of the MAX5215/MAX5217 with R/$\overline{W}$ = 0 to indicate a write. The second byte contains the register (or command) to be written and the third and fourth bytes contain the data to be written. By repeating register address plus data pairs (Byte #2 through Byte #4 in Figure 4 and Figure 5), the user can perform multiple register writes using a single I²C command sequence; there is no limit as to how many registers the user can write with a single command. The MAX5215/MAX5217 support this capability for all user-accessible write mode commands.

I²C Write Operation (Multibyte Protocol)

The MAX5215/MAX5217 support a multibyte transfer protocol for some commands. In multibyte mode, once a command is issued, that command is continuously executed based on two byte data blocks for the duration I²C operation. Essentially, bytes 1 to 4 are processed normally, but for every two bytes of data provided after

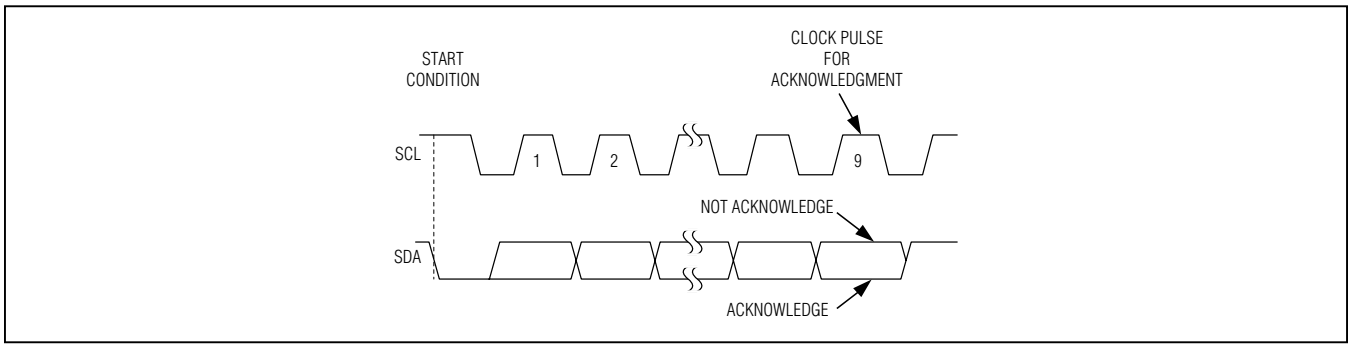


Figure 3. I²C Acknowledge

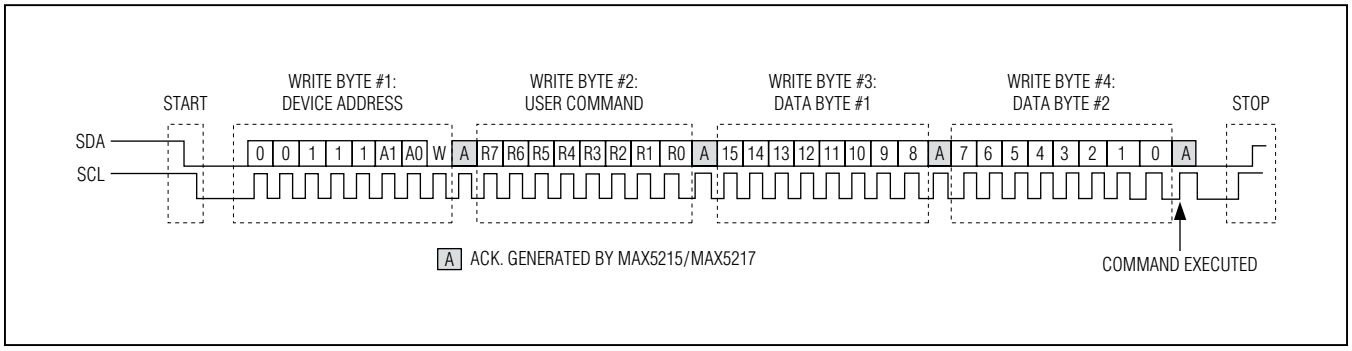


Figure 4. I²C Single Register Write Sequence

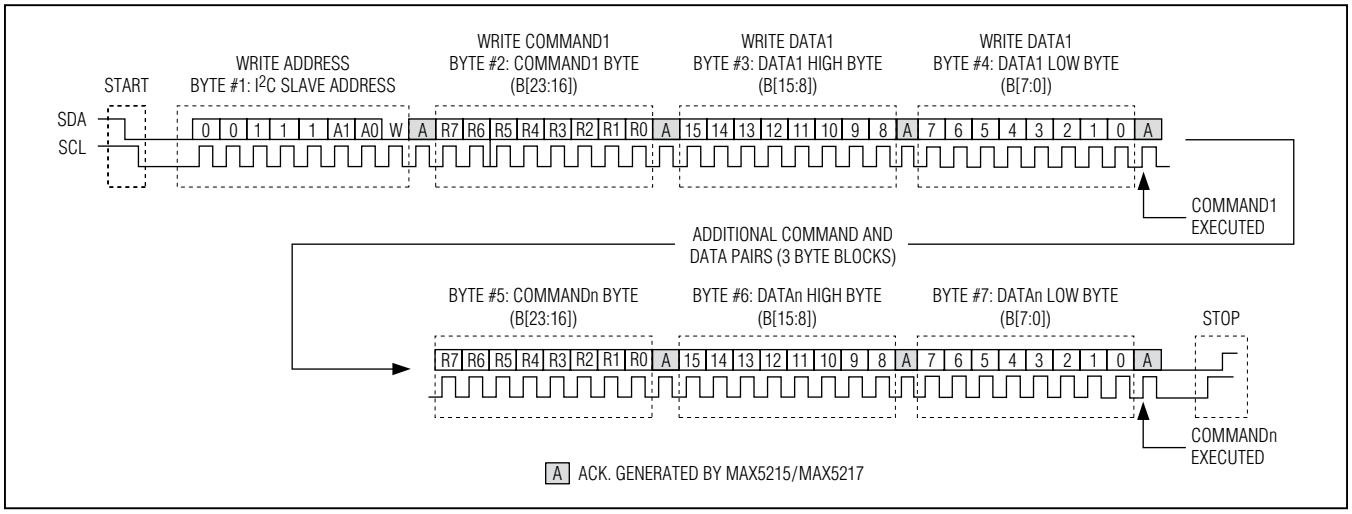


Figure 5. Multiple Register Write Sequence (Standard I²C Protocol)

byte 4, the originally requested command is executed again with the latest byte pair provided as input data (Figure 6). Multibyte protocol is enforced until a STOP condition (or Repeated START) is encountered and this provides a higher speed transfer mode that is useful in servo DAC applications.

I²C Readback Operation

Each readback sequence is framed by a START or Repeated START condition and a STOP condition. Each word is 8 bits long and is followed by an acknowledge clock pulse (Figure 7). The first byte contains the address of the MAX5215/MAX5217 with R/$\overline{W}$ = 0 to indicate a write. The second byte contains the register that is to be read back. There is a Repeated START condition, followed by the device address with R/$\overline{W}$ = 1 to indicate a read and an acknowledge clock. The final two bytes in the frame contain the register data readback followed by a STOP condition. The master has control of the SCL line but the MAX5215/MAX5217 take over the SDA line. Following each byte of data read back from the MAX5215/MAX5217 the master must acknowledge the transfer by pulling SDA low. If additional bytes beyond those required to read back the requested data are provided, the MAX5215/5217 will continue to read back ones.

A user can read back the device's configuration, Part ID, CODE register, or DAC register contents using the readback programming sequence as shown in Figure 7.

I²C Compatibility

The MAX5215/MAX5217 are fully compatible with existing I²C systems. SCL and SDA are high-impedance inputs; SDA has an open drain which pulls the data line low during the 9th clock pulse or as required for data readback. Figure 8 shows a typical I²C application.

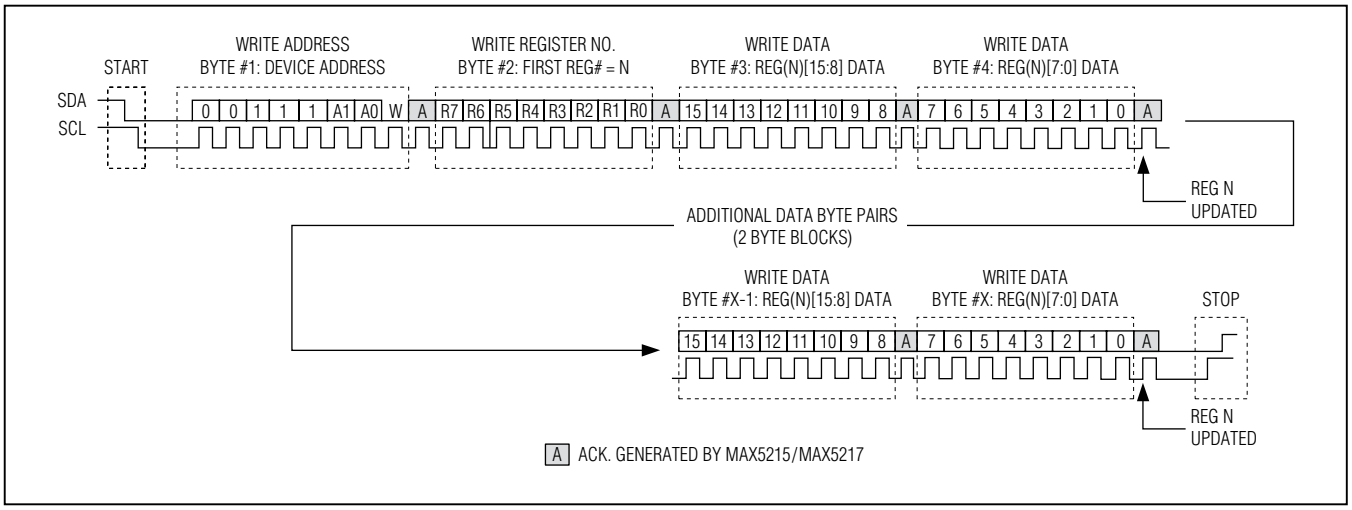


Figure 6. Multiple Register Write Sequence (Multibyte Protocol)

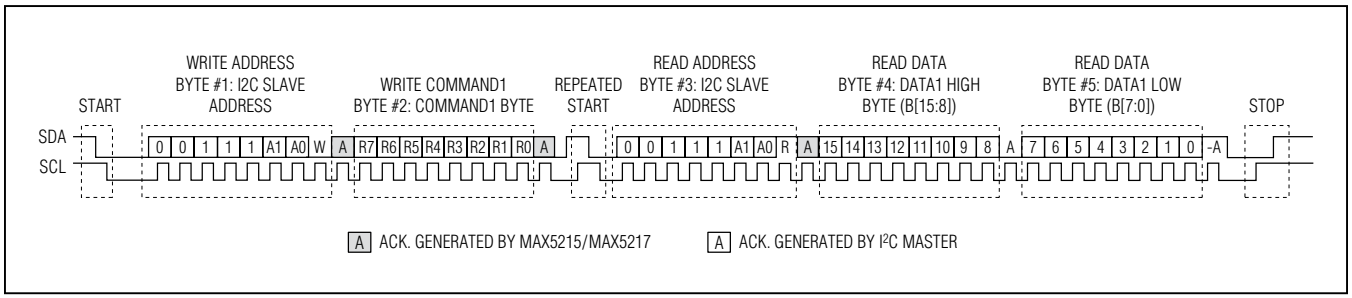


Figure 7. Standard I²C Read Sequence

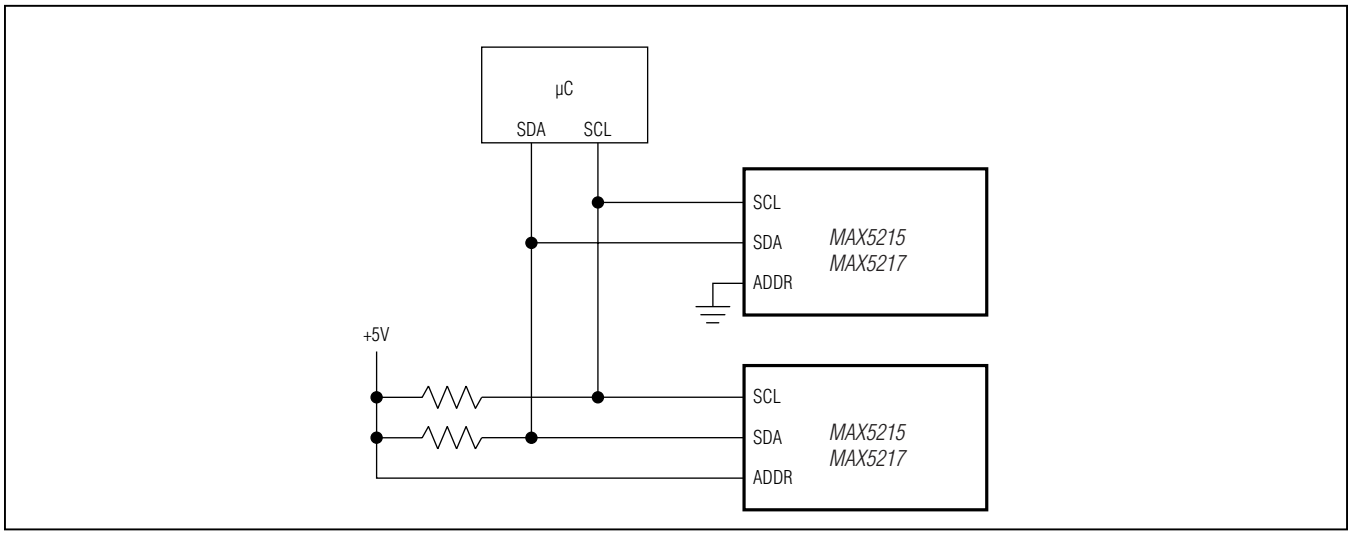


Figure 8. Typical I²C Application Circuit

I²C User-Command Register Map

This section lists the user-accessible commands and registers for the MAX5215/MAX5217. Table 2 provides detailed information about the Command Registers.

No_Op Command (0x00)

The No_Op command (Table 3) has no external effect on the device for I²C write. The asynchronous $\overline{CLR}$ input has no effect on the No_Op command.

Table 2. I²C User Write Commands

WRITE COMMAND	COMMAND BYTE								DATA BYTES	DESCRIPTION	$\overline{CLR}$ GATED*
	R7	R6	R5	R4	R3	R2	R1	R0			
No-Op (0x00)	0	0	0	0	0	0	0	0	Don't Care	No operation: DAC settings and modes unaffected	N
CODE_LOAD (0x01)	0	0	0	0	0	0	0	1	14-/16-bit code	Write and load data to the CODE and DAC registers	Y
CODE (0x02)	0	0	0	0	0	0	1	0	14-/16-bit code	Write data to the CODE register	Y
LOAD (0x03)	0	0	0	0	0	0	1	1	Don't Care	Load current CODE register content to the DAC register	Y
CODE_LOAD_m (0x05)	0	0	0	0	0	1	0	1	Multiple sets of 14-/16-bit codes	Similar to CODE_LOAD command, but accepts multiple sets of dual-byte data following the initial command byte (see the *I²C Write Operation* (*Multibyte Protocol)* section)	Y
CODE_m (0x06)	0	0	0	0	0	1	1	0	Multiple sets of 14-/16-bit codes	Similar to CODE command, but accepts multiple sets of dual-byte data following the initial command byte (see the *I²C Write Operation* (*Multibyte Protocol)* section)	Y
USER_CONFIG (0x08)	0	0	0	0	1	0	0	0	16-bit configuration data	User configuration command	N
SW_RESET (0x09)	0	0	0	0	1	0	0	1	Don't Care	Software Reset	N
SW_CLEAR (0x0A)	0	0	0	0	1	0	1	0	Don't Care	Software Clear	N
Reserved	Any commands not specifically listed above are reserved for Maxim internal use only.										

***Note:** If a user write command is gated by $\overline{CLR}$, and $\overline{CLR}$ has been asserted during the I²C write sequence, the command is ignored and the associated data bytes will not be acknowledged. If a user write command is not gated by $\overline{CLR}$, the command is executed as normal, regardless of the activity of the $\overline{CLR}$ pin.

Table 3. No_Op Command (0x00)

R7	R6	R5	R4	R3	R2	R1	R0	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
0000_0000 No_Op Command								Don't Care								Don't Care							
COMMAND BYTE								DATA HIGH BYTE								DATA LOW BYTE							

CODE_LOAD Command (0x01)

The CODE_LOAD command (Table 4) is the combination of the CODE command and LOAD command. The CODE_LOAD command is executed on the SCL rising edge following the 2nd data byte. Upon its execution, the CODE_LOAD command updates the CODE register and the DAC latch with the user data content provided.

The asynchronous $\overline{CLR}$ input gates this command if it is asserted during the I²C write sequence.

CODE Command (0x02)

The CODE command (Table 5) is executed on the SCL rising edge following the 2nd data byte. The CODE command updates the CODE register with the user data content provided.

The asynchronous $\overline{CLR}$ input gates this command if it is asserted during the I²C write sequence.

LOAD Command (0x03)

The LOAD command (Table 6) is executed on the SCL rising edge following the 2nd data byte. The LOAD command loads the DAC latches with the current contents of the CODE register. Alternatively, a load operation can be achieved by driving the $\overline{AUX}$ input low (when configured as $\overline{LDAC}$).

The asynchronous $\overline{CLR}$ input gates this command if it is asserted during the I²C write sequence.

CODE_LOAD_m Command (0x05)

The CODE_LOAD_m command (Table 7) is the multibyte version of the CODE_LOAD command. The CODE_LOAD_m command is initially executed on the SCL rising edge following the 2nd data byte. The command is subsequently executed after each pair of data bytes which follow, for the duration of the operation (see the *I²C Write Operation (Multibyte Protocol)* section).

The asynchronous $\overline{CLR}$ input gates this command if it is asserted during the I²C write sequence.

Table 4. CODE_LOAD Command

R7	R6	R5	R4	R3	R2	R1	R0	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
0000_0001 16-bit CODE_LOAD Command								CODE and DAC Registers Data								CODE and DAC Registers Data							
0000_0001 14-bit CODE_LOAD Command								CODE and DAC Registers Data								CODE and DAC Registers Data						Don't Care	
COMMAND BYTE								DATA HIGH BYTE								DATA LOW BYTE							

Table 5. CODE Command

R7	R6	R5	R4	R3	R2	R1	R0	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
0000_0010 16-bit CODE Command								CODE Register Data								CODE Register Data							
0000_0010 14-bit CODE Command								CODE Register Data								CODE Register Data						Don't Care	
COMMAND BYTE								DATA HIGH BYTE								DATA LOW BYTE							

Table 6. Load Command

R7	R6	R5	R4	R3	R2	R1	R0	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
0000_0011 LOAD Command								Don't Care								Don't Care							
COMMAND BYTE								DATA HIGH BYTE								DATA LOW BYTE							

CODE_m Command (0x06)

The CODE_m command (Table 8) is the multibyte version of the CODE command. The CODE_m command is initially executed on the SCL rising edge following the 2nd data byte. The command is subsequently executed after each pair of data bytes which follow, for the duration of the operation (see the *I²C Write Operation (Multibyte Protocol)* section). This command is of practical use when the $\overline{AUX}$ pin is configured as $\overline{LDAC}$ and continuously asserted low.

The asynchronous $\overline{CLR}$ input gates this command if it is asserted during the I²C write sequence.

USER_CONFIG Command (0x08)

The USER_CONFIG command allows the user to select the configuration of the device: setting the clear value to which the DAC returns in response to a CLEAR event, configuring the input mode for $\overline{AUX}$, and setting the power-down mode for the MAX5215/MAX5217. The USER_CONFIG command is executed on the SCL rising edge following the 2nd data byte. Table 9 and Table 10 describe the command and the configuration bits in detail.

The asynchronous $\overline{CLR}$ input has no effect on the USER_CONFIG command.

Table 7. CODE_LOAD_m Command

R7	R6	R5	R4	R3	R2	R1	R0	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
0000_0101 16-bit CODE_LOAD_m Command								CODE and DAC Registers Data								CODE and DAC Registers Data							
0000_0101 14-bit CODE_LOAD_m Command								CODE and DAC Registers Data								CODE and DAC Registers Data						Don't Care	
COMMAND BYTE								DATA HIGH BYTE								DATA LOW BYTE							

Table 8. CODE_m Command

R7	R6	R5	R4	R3	R2	R1	R0	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
0000_0110 16-bit CODE_m Command								CODE and DAC Registers Data								CODE and DAC Registers Data							
0000_0110 14-bit CODE_m Command								CODE and DAC Registers Data								CODE and DAC Registers Data						Don't Care	
COMMAND BYTE								DATA HIGH BYTE								DATA LOW BYTE							

Table 9. USER_CONFIG Command

R7	R6	R5	R4	R3	R2	R1	R0	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
0000_1000 USER_CONFIG Command								Don't Care								Don't Care		**Clear Value Mode:** 00 = Default 01 = Zero 10 = Mid 11 = Full		**$\overline{AUX}$ Input Mode:** 00 = Disable 01 = $\overline{LDAC}$ 10 = $\overline{CLR}$ 11 = Disable		**Power-Down Mode:** 00 = DAC 01 = High-Z 10 = 100kΩ 11 = 1kΩ	
DATA DEFAULT VALUE→								X	X	X	X	X	X	X	X	X	X	0	0	1	0	0	0
COMMAND BYTE								DATA HIGH BYTE								DATA LOW BYTE							

SW_RESET Command (0x09)

The SW_RESET command (Table 11) resets the CODE register, the DAC latch, and all the configurations programmed via the USER_CONFIG command to the POR default values. The SW_RESET command is executed on the SCL rising edge following the second data byte.

The asynchronous $\overline{CLR}$ input has no effect on the SW_RESET command.

SW_CLEAR Command (0x0A)

The SW_CLEAR command (Table 12) will clear the CODE register and the DAC latch to the clear value selected in the USER_CONFIG register. The SW_CLEAR command is executed on the SCL rising edge following the 2nd data byte. Alternatively, a clear operation can be achieved by driving the $\overline{AUX}$ input low (when configured as $\overline{CLR}$).

The asynchronous $\overline{CLR}$ input has no effect on the SW_CLEAR command.

Table 10. User_Configuration Bits (B[5:0])

CONFIGURATION BITS	CONFIGURATION DETAIL
CLEAR VALUE (B[5:4])	The DAC value to be cleared to in response to a CLEAR event: 00: POR default value (zero scale) 01: Zero scale (ground) 10: Midscale 11: Full scale (reference)
$\overline{AUX}$ MODE (B[3:2])	The mode in which the $\overline{AUX}$ input will operate: 00: Pin disabled 01: Enable $\overline{LDAC}$ functionality 10: Enable $\overline{CLR}$ functionality. Default after POR. 11: Pin disabled
Power-Down Mode (PD) (B[1:0])	Power-down mode for the device: 00: Normal operation: The DAC will be powered up and returned to its previous setting. Default after POR. 01: Power-down: The DAC core will be powered down and V_{OUT} is high-impedance. 10: Power-down: The DAC core will be powered down and V_{OUT} is connected to ground via 100kΩ. 11: Power-down: The DAC core will be powered down and V_{OUT} is connected to ground via 1kΩ.

Table 11. SW_RESET Command

R7	R6	R5	R4	R3	R2	R1	R0	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
0000_1001 SW_RESET Command								Don't Care								Don't Care							
COMMAND BYTE								DATA HIGH BYTE								DATA LOW BYTE							

Table 12. SW_CLEAR Command

R7	R6	R5	R4	R3	R2	R1	R0	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
0000_1010 SW_CLEAR Command								Don't Care								Don't Care							
COMMAND BYTE								DATA HIGH BYTE								DATA LOW BYTE							

User Read Command Descriptions

The MAX5215/MAX5217 allow the user to read back the data for supported registers. Table 13 lists the User Readback commands and the 2 data byte contents being read back. For the MAX5215, CODE and DAC read back, the data content is left justified and the 2 LSBs ([1:0]) of the input 2-byte data are not used and read out as 0.

Applications Information

Power-On Reset (POR)

When power is applied to V_{DD}, the input registers are set to zero so the DAC output is set to code zero. Initially the device powers up to an untrimmed zero code setting. The device will operate in a fully trimmed mode following the first I²C operation which modifies DAC latch content.

Power Supplies and Bypassing Consideations

Bypass V_{DD} with high-quality ceramic capacitors to a low-impedance ground as close as possible to the device. Minimize lead lengths to reduce lead inductance. Connect the GND input to the analog ground plane.

Layout Considerations

Digital and AC transient signals on GND can create noise at the output. Connect GND to form the star ground for the DAC system. Refer remote DAC loads to this system ground for the best possible performance. Use proper grounding techniques, such as a multilayer board with a low-inductance ground plane, or star connect all ground return paths back to the MAX5215/MAX5217 GND. Carefully lay out the traces to reduce AC cross-coupling. Do not use wire-wrapped boards and sockets. Use shielding to improve noise immunity. Do not run analog and digital signals parallel to one another, especially clock signals. Avoid routing digital lines underneath the MAX5215/MAX5217 package.

Table 13. User Readback Command and Content

R7	R6	R5	R4	R3	R2	R1	R0	READ COMMAND	READ DATA1 HIGH BYTE D[15:8]	READ DATA1 LOW BYTE D[7:0]
0	0	0	0	0	0	0	0	ID Readback (0x00)	0011100, CLEAR Status	(0x11)
0	0	0	0	0	0	0	1	CODE_LOAD Readback (0x01)	DAC_latch[15:8]	DAC_latch[7:0]
0	0	0	0	0	0	1	0	CODE Readback (0x02)	CODE_register[15:8]	CODE_register[7:0]
0	0	0	0	0	0	1	1	LOAD Readback (0x03)	DAC_latch[15:8]	DAC_latch[7:0]
0	0	0	0	0	1	0	1	CODE_LOAD_m Readback (0x05)	DAC_latch[15:8]	DAC_latch[7:0]
0	0	0	0	0	1	1	0	CODE_m Readback (0x06)	CODE_register[15:8]	CODE_register[7:0]
0	0	0	0	1	0	0	0	CONFIG Readback (0x08)	0000_0000	00, CLEAR_VALUE[1:0], $\overline{\text{AUX}}$_MODE[1:0], PD[1:0]

Definitions

Integral Nonlinearity (INL)

INL is the deviation of the measured transfer function from a straight line drawn between two codes once offset and gain errors have been nullified.

Differential Nonlinearity (DNL)

DNL is the difference between an actual step height and the ideal value of 1 LSB. If the magnitude of the DNL is greater than -1 LSB, the DAC guarantees no missing codes and is monotonic.

Offset Error

Offset error indicates how well the actual transfer function matches the ideal transfer function. The offset error is calculated from two measurements near zero code and near maximum code.

Gain Error

Gain error is the difference between the ideal and the actual full-scale output voltage on the transfer curve, after nullifying the offset error. This error alters the slope of the transfer function and corresponds to the same percentage error in each step.

Settling Time

The settling time is the amount of time required from the start of a transition, until the DAC output settles to the new output value within the measurment's specified accuracy.

Digital Feedthrough

Digital feedthrough is the amount of noise that appears on the DAC output when the DAC digital control lines are toggled.

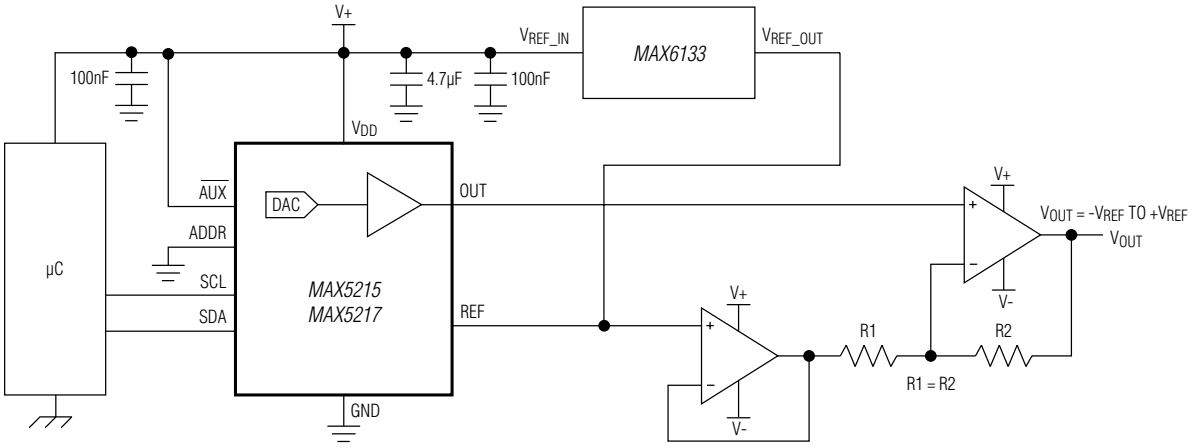
Digital-to-Analog Glitch Impulse

A major carry transition occurs at the midscale point where the MSB changes from low to high and all other bits change from high to low, or where the MSB changes from high to low and all other bits change from low to high. The duration of the magnitude of the switching glitch during a major carry transition is referred to as the digital-to-analog glitch impulse.

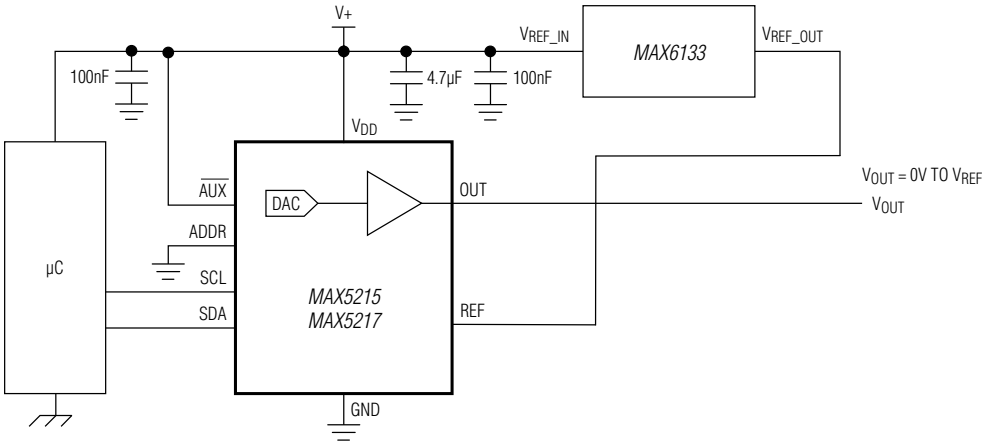
Digital-to-Analog Power-Up Glitch Impulse

The digital-to-analog power-up glitch is the duration of the magnitude of the switching glitch that occurs as the device exits power-down mode.

Typical Operating Circuit



BIPOLAR OPERATION



UNIPOLAR OPERATION

Ordering Information

PART	PIN-PACKAGE	RESOLUTION (BITS)	INL MAX (LSB)
MAX5215GUA+	8 µMAX	14	±1
MAX5217GUA+	8 µMAX	16	±4
MAX5217BGUA+	8 µMAX	16	±8

Note: *All devices are specified over the -40°C to +105°C operating temperature range.*
+Denotes a lead(Pb)-free/RoHS-compliant package.
R = Tape and reel.

Chip Information

PROCESS: BiCMOS

Package Information

For the latest package outline information and land patterns (footprints), go to **www.maximintegrated.com/packages**. Note that a "+", "#", or "-" in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

PACKAGE TYPE	PACKAGE CODE	OUTLINE NO.	LAND PATTERN NO.
8 µMAX	U8+3	21-0036	90-0092

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	11/12	Initial release	—

Maxim Integrated cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim Integrated product. No circuit patent licenses are implied. Maxim Integrated reserves the right to change the circuitry and specifications without notice at any time. The parametric values (min and max limits) shown in the Electrical Characteristics table are guaranteed. Other parametric values quoted in this data sheet are provided for guidance.

© 2012 Maxim Integrated Maxim Integrated and the Maxim Integrated logo are trademarks of Maxim Integrated Products, Inc.