



Octal 8-Bit TrimDAC with Power Shutdown

AD8801/AD8803

FEATURES

- Low Cost
- Replaces Eight Potentiometers
- Eight Individually Programmable Outputs
- Three-Wire Serial Input
- Power Shutdown $\leq 25 \mu\text{W}$ Including I_{DD} and I_{REF}
- Midscale Preset, AD8801
- Separate V_{REFL} Range Setting, AD8803
- +3 V to +5 V Single Supply Operation

APPLICATIONS

- Automatic Adjustment
- Trimmer Potentiometer Replacement
- Video and Audio Equipment Gain and Offset Adjustment
- Portable and Battery Operated Equipment

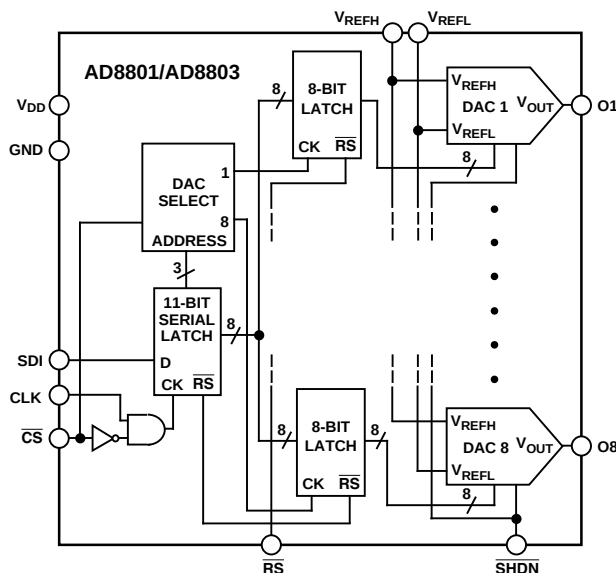
GENERAL DESCRIPTION

The AD8801/AD8803 provides eight digitally controlled dc voltage outputs. This potentiometer divider TrimDAC[®] allows replacement of the mechanical trimmer function in new designs. The AD8801/AD8803 is ideal for dc voltage adjustment applications.

Easily programmed by serial interfaced microcontroller ports, the AD8801 with its midscale preset is ideal for potentiometer replacement where adjustments start at a nominal value. Applications such as gain control of video amplifiers, voltage controlled frequencies and bandwidths in video equipment, geometric correction and automatic adjustment in CRT computer graphic displays are a few of the many applications ideally suited for these parts. The AD8803 provides independent control of both the top and bottom end of the potentiometer divider allowing a separate zero-scale voltage setting determined by the V_{REFL} pin. This is helpful for maximizing the resolution of devices with a limited allowable voltage control range.

FUNCTIONAL BLOCK DIAGRAM

(DACs 2–7 Omitted for Clarity)



Internally the AD8801/AD8803 contain eight voltage output digital-to-analog converters, sharing a common reference voltage input.

Each DAC has its own DAC register that holds its output state. These DAC registers are updated from an internal serial-to-parallel shift register that is loaded from a standard three-wire serial input digital interface. Eleven data bits make up the data word clocked into the serial input register. This data word is decoded where the first 3 bits determine the address of the DAC register to be loaded with the last 8 bits of data. The AD8801/AD8803 consumes only $5 \mu\text{A}$ from 5 V power supplies. In addition, in shutdown mode reference input current consumption is also reduced to $5 \mu\text{A}$ while saving the DAC latch settings for use after return to normal operation.

The AD8801/AD8803 is available in 16-pin plastic DIP and the 1.5 mm height SO-16 surface mount packages.

See the AD8802/AD8804 for a twelve channel version of this product.

TrimDAC is a registered trademark of Analog Devices, Inc.

REV. A

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

© Analog Devices, Inc., 1995

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 617/329-4700 Fax: 617/326-8703

AD8801/AD8803—SPECIFICATIONS ($V_{DD} = +3\text{ V} \pm 10\%$ or $+5\text{ V} \pm 10\%$, $V_{REFH} = +V_{DD}$, $V_{REFL} = 0\text{ V}$, $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ unless otherwise noted)

Parameter	Symbol	Conditions	Min	Typ ¹	Max	Units
STATIC ACCURACY						
Specifications Apply to All DACs						
Resolution	N	Guaranteed Monotonic	8			Bits
Integral Nonlinearity Error	INL		−1.5	±1/2	+1.5	LSB
Differential Nonlinearity	DNL		−1	±1/4	+1	LSB
Full-Scale Error	G _{FSE}		−4	−2.8	+0.5	LSB
Zero-Code Error	V _{ZSE}		−0.5	±0.1	+0.5	LSB
DAC Output Resistance	R _{OUT}		3	5	8	kΩ
Output Resistance Match	ΔR/R _O				1	%
REFERENCE INPUT						
Voltage Range ²	V _{REFH} V _{REFL}	Pin Available on AD8803 Only Digital Inputs = 55 _H , V _{REFH} = V _{DD} Digital Inputs All Zeros Digital Inputs All Ones	0 0		V _{DD} V _{DD}	V V
Input Resistance	R _{REFH}			2		kΩ
Reference Input Capacitance ³	C _{REF0}			25		pF
	C _{REF1}			25		pF
DIGITAL INPUTS						
Logic High	V _{IH}	V _{DD} = +5 V	2.4			V
Logic Low	V _{IL}	V _{DD} = +5 V			0.8	V
Logic High	V _{IH}	V _{DD} = +3 V	2.1			V
Logic Low	V _{IL}	V _{DD} = +3 V			0.6	V
Input Current	I _{IL}	V _{IN} = 0 V or +5 V			±1	μA
Input Capacitance ³	C _{IL}			5		pF
POWER SUPPLIES ⁴						
Power Supply Range	V _{DD} Range		2.7		5.5	V
Supply Current (CMOS)	I _{DD}	V _{IH} = V _{DD} or V _{IL} = 0 V		0.01	5	μA
Supply Current (TTL)	I _{DD}	V _{IH} = 2.4 V or V _{IL} = 0.8 V, V _{DD} = +5.5 V		1	4	mA
Shutdown Current	I _{REFH}	$\overline{\text{SHDN}}$ = 0		0.01	5	μA
Power Dissipation	P _{DISS}	V _{IH} = V _{DD} or V _{IL} = 0 V, V _{DD} = +5.5 V			27.5	μW
Power Supply Sensitivity	PSRR	V _{DD} = 5 V ± 10%, V _{REFH} = +4.5 V		0.001	0.002	%/%
Power Supply Sensitivity	PSRR	V _{DD} = 3 V ± 10%, V _{REFH} = +2.7 V		0.01		%/%
DYNAMIC PERFORMANCE ³						
V _{OUT} Settling Time (Positive or Negative)	t _S	±1/2 LSB Error Band		0.6		μs
Crosstalk	CT	See Note 5, f = 100 kHz		50		dB
SWITCHING CHARACTERISTICS ^{3, 6}						
Input Clock Pulse Width	t _{CH} , t _{CL}	Clock Level High or Low	15			ns
Data Setup Time	t _{DS}		5			ns
Data Hold Time	t _{DH}		5			ns
$\overline{\text{CS}}$ Setup Time	t _{CSS}		10			ns
$\overline{\text{CS}}$ High Pulse Width	t _{CSW}		10			ns
Reset Pulse Width	t _{RS}		60			ns
CLK Rise to $\overline{\text{CS}}$ Rise Hold Time	t _{CSH}		15			ns
$\overline{\text{CS}}$ Rise to Next Rising Clock	t _{CS1}		10			ns

NOTES

¹Typical values represent average readings measured at +25°C.

² V_{REFH} can be any value between GND and V_{DD} , for the AD8803 V_{REFL} can be any value between GND and V_{DD} .

³Guaranteed by design and not subject to production test.

⁴Digital Input voltages $V_{IN} = 0\text{ V}$ or V_{DD} for CMOS condition. DAC outputs unloaded. P_{DISS} is calculated from $(I_{DD} \times V_{DD})$.

⁵Measured at a V_{OUT} pin where an adjacent V_{OUT} pin is making a full-scale voltage change.

⁶See timing diagram for location of measured values. All input control voltages are specified with $t_R = t_F = 2\text{ ns}$ (10% to 90% of V_{DD}) and timed from a voltage level of 1.6 V.

Specifications subject to change without notice.

ABSOLUTE MAXIMUM RATINGS(T_A = +25°C, unless otherwise noted)

V _{DD} to GND	−0.3, +8 V
V _{REFX} to GND	0 V, V _{DD}
Outputs (Ox) to GND	0 V, V _{DD}
Digital Input Voltage to GND	0 V, V _{DD}
Operating Temperature Range	−40°C to +85°C
Maximum Junction Temperature (T _J MAX)	+150°C
Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 sec)	+300°C
Package Power Dissipation	(T _J MAX − T _A)/θ _{JA}
Thermal Resistance θ _{JA} ,	
SOIC (SO-16)	60°C/W
P-DIP (N-16)	57°C/W

AD8801 PIN DESCRIPTIONS

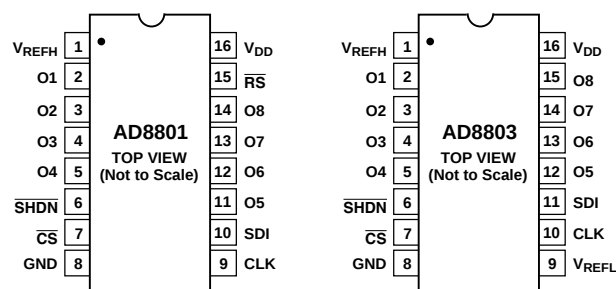
Pin	Name	Description
1	V _{REFH}	Common DAC Reference Input
2	O1	DAC Output #1, Addr = 000 ₂
3	O2	DAC Output #2, Addr = 001 ₂
4	O3	DAC Output #3, Addr = 010 ₂
5	O4	DAC Output #4, Addr = 011 ₂
6	SHDN	Reference input open circuit, active low, all DAC outputs open circuit. DAC latch settings maintained.
7	CS	Chip Select Input, active low. When CS returns high, data in the serial input register is decoded based on the address bits and loaded into the target DAC register.
8	GND	Ground
9	CLK	Serial Clock Input, Positive Edge Triggered
10	SDI	Serial Data Input
11	O5	DAC Output #5, Addr = 100 ₂
12	O6	DAC Output #6, Addr = 101 ₂
13	O7	DAC Output #7, Addr = 110 ₂
14	O8	DAC Output #8, Addr = 111 ₂
15	RS	Asynchronous preset to midscale output setting, active low. Loads all DAC latches with 80 _H .
16	V _{DD}	Positive power supply, specified for operation at both +3 V and +5 V.

ORDERING GUIDE

Model	FTN	Temperature	Package Description	Package Option
AD8801AN	RS	−40°C to +85°C	PDIP-16	N-16
AD8801AR	RS	−40°C to +85°C	SO-16	R-16A
AD8803AN	REFL	−40°C to +85°C	PDIP-16	N-16
AD8803AR	REFL	−40°C to +85°C	SO-16	R-16A

AD8803 PIN DESCRIPTIONS

Pin	Name	Description
1	V _{REFH}	Common High-Side DAC Reference Input
2	O1	DAC Output #1, Addr = 000 ₂
3	O2	DAC Output #2, Addr = 001 ₂
4	O3	DAC Output #3, Addr = 010 ₂
5	O4	DAC Output #4, Addr = 011 ₂
6	SHDN	Reference inputs open circuit, active low, all DAC outputs open circuit. DAC latch settings maintained.
7	CS	Chip Select Input, active low. When CS returns high, data in the serial input register is decoded based on the address bits and loaded into the target DAC register.
8	GND	Ground
9	V _{REFL}	Common Low-Side DAC Reference Input
10	CLK	Serial Clock Input, Positive Edge Triggered
11	SDI	Serial Data Input
12	O5	DAC Output #5, Addr = 100 ₂
13	O6	DAC Output #6, Addr = 101 ₂
14	O7	DAC Output #7, Addr = 110 ₂
15	O8	DAC Output #8, Addr = 111 ₂
16	V _{DD}	Positive power supply, specified for operation at both +3 V and +5 V.

PIN CONFIGURATIONS**CAUTION**

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although these devices feature proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.



AD8801/AD8803

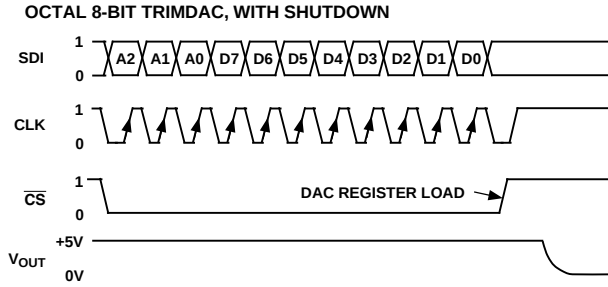


Figure 2a. Timing Diagram

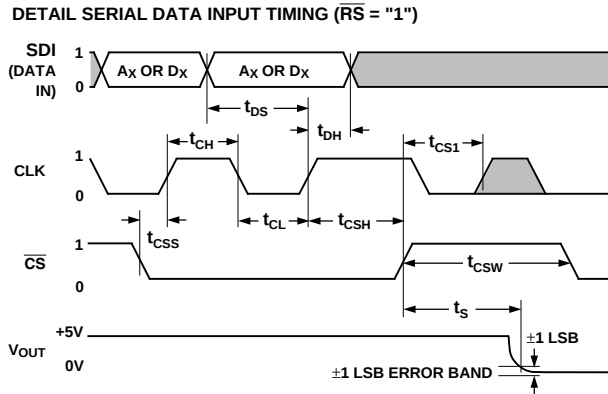


Figure 2b. Detail Timing Diagram

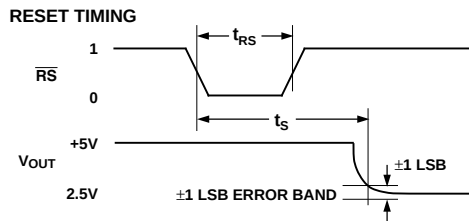


Figure 2c. Reset Timing Diagram

Table I. Serial-Data Word Format

ADDR			DATA							
B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
MSB		LSB	MSB							LSB
2 ¹⁰	2 ⁹	2 ⁸	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰

OPERATION

The AD8801/AD8803 provides eight channels of programmable voltage output adjustment capability. Changing the programmed output voltage of each TrimDAC is accomplished by clocking in an 11-bit serial data word into the SDI (Serial Data Input) pin. The format of this data word is three address bits, MSB first, followed by eight data bits, MSB first. Table I provides the serial register data word format. The AD8801/AD8803 has the

following address assignments for the ADDR decode which determines the location of DAC register receiving the serial register data in bits B7 through B0:

$$DAC \# = A2 \times 4 + A1 \times 2 + A0 + 1$$

DAC outputs can be changed one at a time in random sequence. The fast serial-data loading of 33 MHz makes it possible to load all eight DACs in as little time as 3 μ s (12 $\times$ 8 $\times$ 30 ns). The exact timing requirements are shown in Figure 2.

The AD8801 offers a midscale preset activated by the $\overline{RS}$ pin simplifying initial setting conditions at first power up. The AD8803 has both a V_{REFH} and a V_{REFL} pin to establish independent positive full-scale and zero-scale settings to optimize resolution. Both parts offer a power shutdown $\overline{SHDN}$ that places the DAC structure in a zero power consumption state resulting in only leakage currents being consumed from the power supply, V_{REF} inputs, and all 8 outputs. In shutdown mode the DACx latch settings are maintained. When returning to operational mode from power shutdown the DAC outputs return to their previous voltage settings.

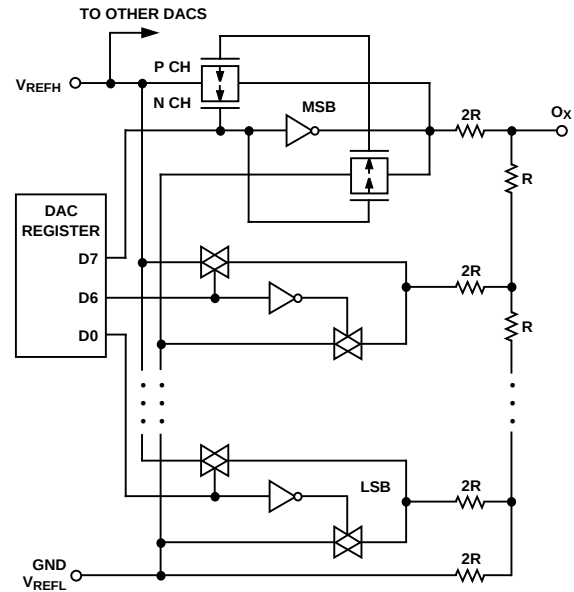


Figure 3. AD8801/AD8803 Equivalent TrimDAC Circuit

PROGRAMMING THE OUTPUT VOLTAGE

The output voltage range is determined by the external reference connected to V_{REFH} and V_{REFL} pins. See Figure 3 for a simplified diagram of the equivalent DAC circuit. In the case of the AD8801, its V_{REFL} is internally connected to GND and therefore cannot be offset. V_{REFH} can be tied to V_{DD} and V_{REFL} can be tied to GND establishing a basic rail-to-rail voltage output programming range. Other output ranges are established by the use of different external voltage references. The general transfer equation that determines the programmed output voltage is:

$$V_O(Dx) = (Dx)/256 \times (V_{REFH} - V_{REFL}) + V_{REFL} \quad (1)$$

where Dx is the data contained in the 8-bit DACx latch.

For example, when $V_{REFH} = +5\text{ V}$ and $V_{REFL} = 0\text{ V}$ the following output voltages will be generated for the following codes:

D	V_{OX}	Output State ($V_{REFH} = +5\text{ V}$, $V_{REFL} = 0\text{ V}$)
255	4.98 V	Full-Scale
128	2.50 V	Half-Scale (Midscale Reset Value)
1	0.02 V	1 LSB
0	0.00 V	Zero-Scale

REFERENCE INPUTS (V_{REFH} , V_{REFL})

The reference input pins set the output voltage range of all eight DACs. In the case of the AD8801 only the V_{REFH} pin is available to establish a user designed full-scale output voltage. The external reference voltage can be any value between 0 and V_{DD} but must not exceed the V_{DD} supply voltage. In the case of the AD8803, which has access to the V_{REFL} which establishes the zero-scale output voltage, any voltage can be applied between 0 V and V_{DD} . V_{REFL} can be smaller or larger in voltage than V_{REFH} since the DAC design uses fully bidirectional switches as shown in Figure 3. The input resistance to the DAC has a code dependent variation that has a nominal worst case measured at 55_H , which is approximately 2 k Ω . When V_{REFH} is greater than V_{REFL} , the REFL reference must be able to sink current out of the DAC ladder, while the REFH reference is sourcing current into the DAC ladder. The DAC design minimizes reference glitch current maintaining minimum interference between DAC channels during code changes.

DAC OUTPUTS (O1–O8)

The eight DAC outputs present a constant output resistance of approximately 5 k Ω independent of code setting. The distribution of R_{OUT} from DAC to DAC typically matches within $\pm 1\%$. However, device to device matching is process lot dependent having a $\pm 20\%$ variation. The change in R_{OUT} with temperature has a 500 ppm/ $^{\circ}\text{C}$ temperature coefficient. During power shut-down all eight outputs are open circuited.

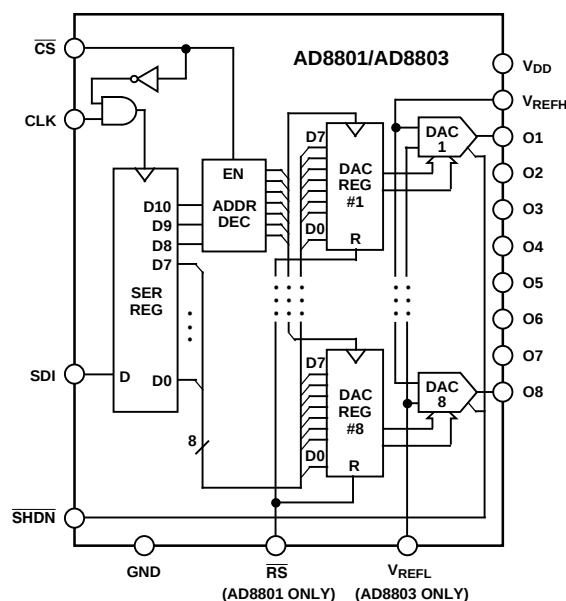


Figure 4. Block Diagram

DIGITAL INTERFACING

The AD8801/AD8803 contains a standard three-wire serial input control interface. The three inputs are clock (CLK), $\overline{\text{CS}}$ and serial data input (SDI). The positive-edge sensitive CLK input requires clean transitions to avoid clocking incorrect data into the serial input register. Standard logic families work well. If mechanical switches are used for product evaluation, they should be debounced by a flip-flop or other suitable means. Figure 4 block diagram shows more detail of the internal digital circuitry. When $\overline{\text{CS}}$ is taken active low, the clock can load data into the serial register on each positive clock edge, see Table II.

Table II. Input Logic Control Truth Table

$\overline{\text{CS}}$	CLK	Register Activity
1	X	No effect.
0	P	Shifts Serial Register one bit loading the next bit in from the SDI pin.
P	X	Data is transferred from the serial register to the decoded DAC register. See Figure 5.

NOTE: P = positive edge, X = don't care.

The data setup and data hold times in the specification table determine the data valid time requirements. The last 11 bits of the data word entered into the serial register are held when $\overline{\text{CS}}$ returns high. At the same time $\overline{\text{CS}}$ goes high it gates the address decoder which enables one of the eight positive edge triggered DAC registers, see Figure 5 detail.

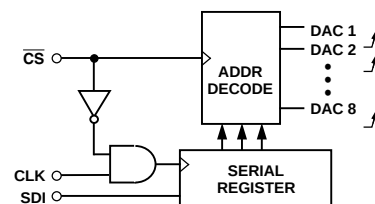


Figure 5. Equivalent Control Logic

The target DAC register is loaded with the last eight bits of the serial data word completing one DAC update. Eight separate 11-bit data words must be clocked in to change all eight output settings.

All digital inputs are protected with a series input resistor and parallel Zener ESD structure shown in Figure 6. This applies to digital input pins $\overline{\text{CS}}$, SDI, $\overline{\text{RS}}$, $\overline{\text{SHDN}}$, CLK.

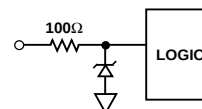


Figure 6. Equivalent ESD Protection Circuit

Digital inputs can be driven by voltages exceeding the AD8801/AD8803 V_{DD} value. This allows 5 V logic to interface directly to the part when it is operated at 3 V.

AD8801/AD8803—Typical Performance Characteristics

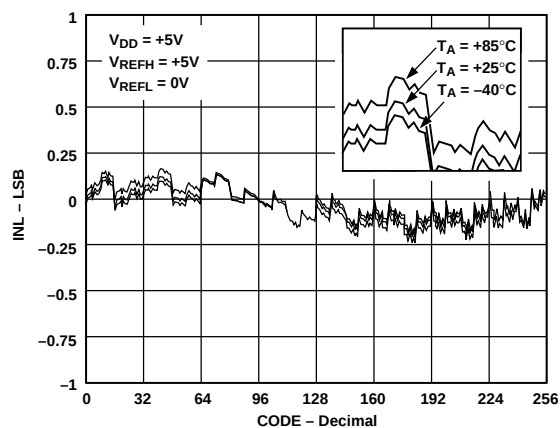


Figure 7. INL vs. Code

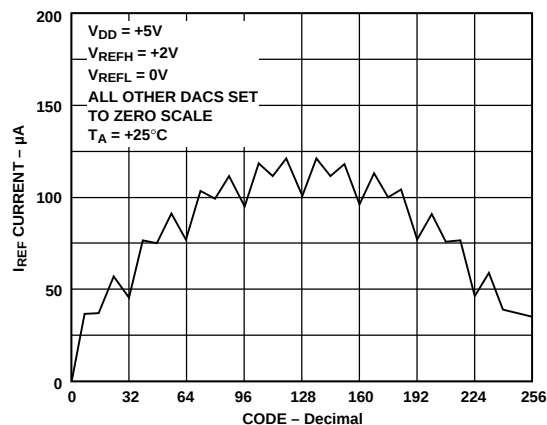


Figure 10. Input Reference Current vs. Code

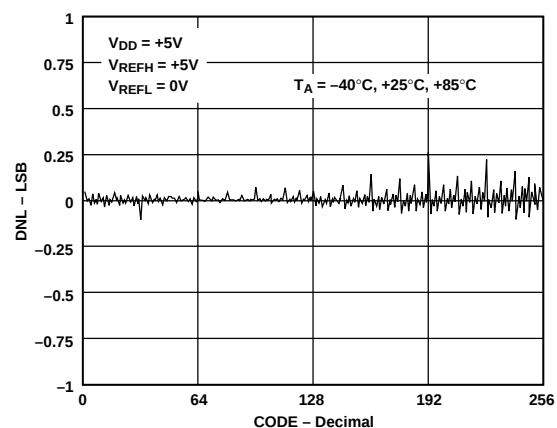


Figure 8. Differential Nonlinearity Error vs. Code

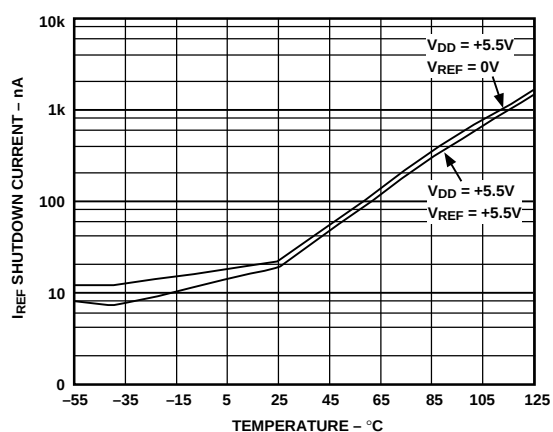


Figure 11. Shutdown Current vs. Temperature

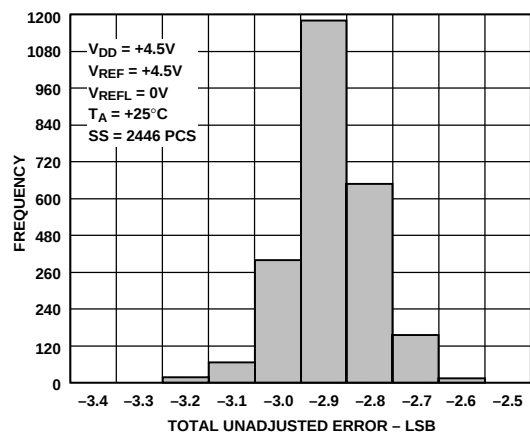


Figure 9. Total Unadjusted Error Histogram

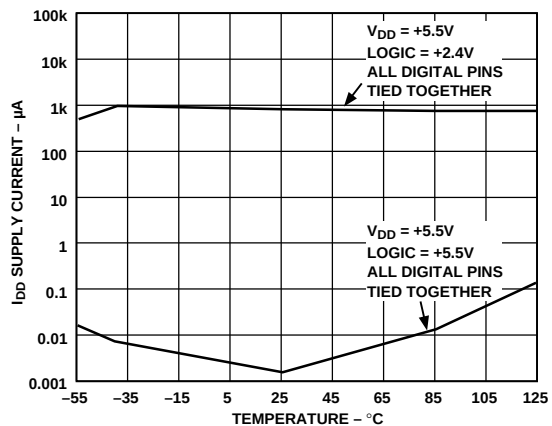


Figure 12. Supply Current vs. Temperature

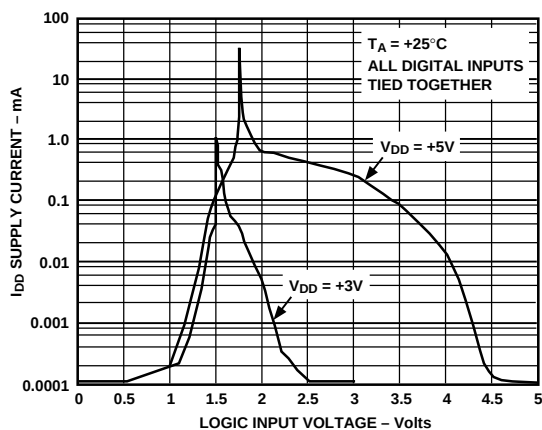


Figure 13. Supply Current vs. Logic Input Voltage

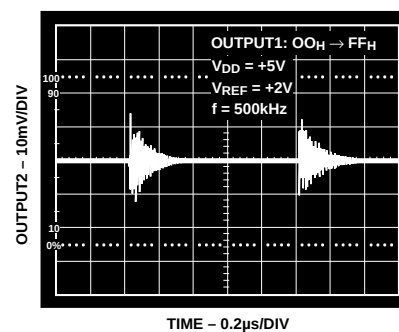


Figure 16. Adjacent Channel Clock Feedthrough

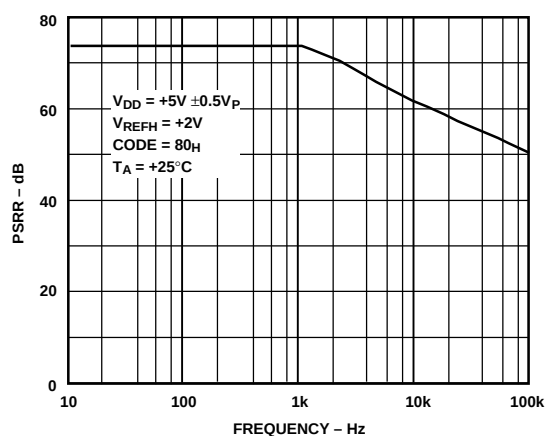


Figure 14. Power Supply Rejection vs. Frequency

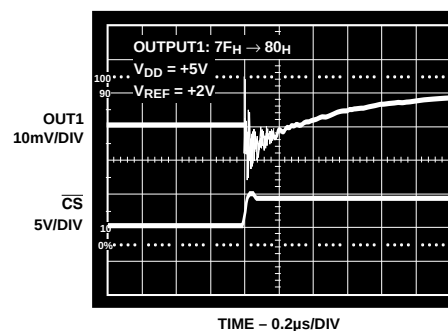


Figure 17. Midscale Transition

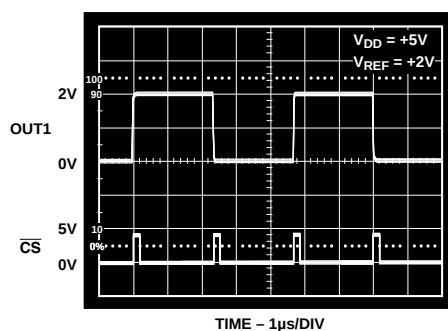


Figure 15. Large-Signal Settling Time

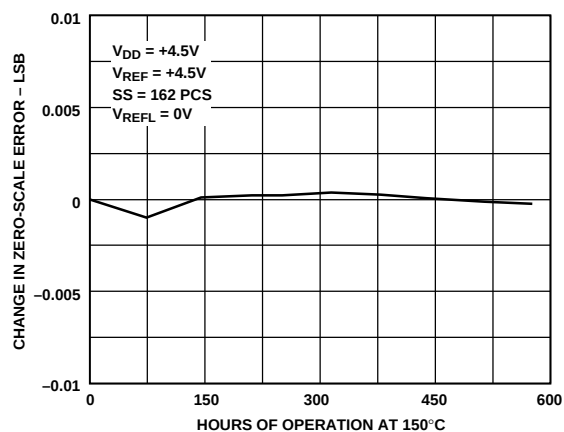


Figure 18. Zero-Scale Error Accelerated by Burn-In

AD8801/AD8803

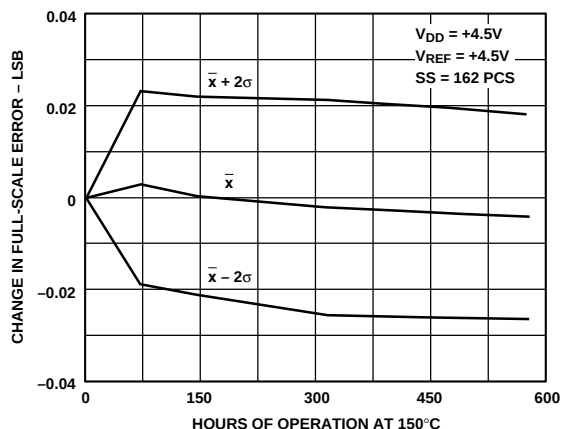


Figure 19. Full-Scale Error Accelerated by Burn-In

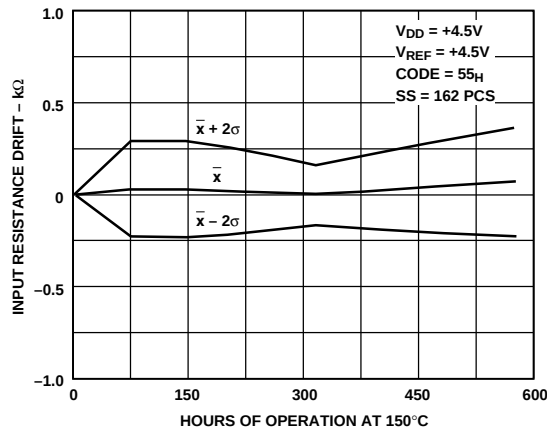


Figure 20. REF Input Resistance Accelerated by Burn-In

APPLICATIONS

Supply Bypassing

Precision analog products, such as the AD8801/AD8803, require a well filtered power source. Since the AD8801/AD8803 operate from a single +3 V to +5 V supply, it seems convenient to simply tap into the digital logic power supply. Unfortunately, the logic supply is often a switch-mode design, which generates noise in the 20 kHz to 1 MHz range. In addition, fast logic gates can generate glitches hundred of millivolts in amplitude due to wiring resistances and inductances.

If possible, the AD8801/AD8803 should be powered directly from the system power supply. This arrangement, shown in Figure 21, will isolate the analog section from the logic switching transients. Even if a separate power supply trace is not available, however, generous supply bypassing will reduce supply-line induced errors. Local supply bypassing consisting of a 10 μ F tantalum electrolytic in parallel with a 0.1 μ F ceramic capacitor is recommended (Figure 22).

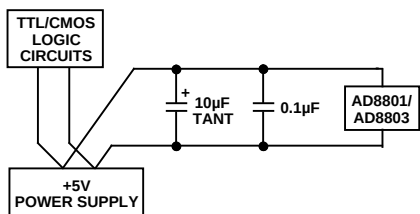


Figure 21. Use Separate Traces to Reduce Power Supply Noise

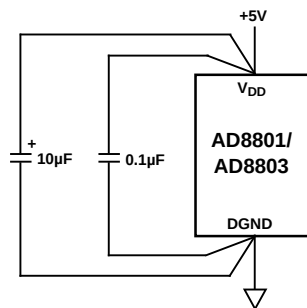


Figure 22. Recommended Supply Bypassing for the AD8801/AD8803

Buffering the AD8801/AD8803 Output

In many cases, the nominal 5 k Ω output impedance of the AD8801/AD8803 is sufficient to drive succeeding circuitry. If a lower output impedance is required, an external amplifier can be added. Several examples are shown in Figure 23. One amplifier of an OP291 is used as a simple buffer to reduce the output resistance of DAC A. The OP291 was chosen primarily for its rail-to-rail input and output operation, but it also offers operation to less than 3 V, low offset voltage, and low supply current.

The next two DACs, B and C, are configured in a summing arrangement where DAC C provides the coarse output voltage setting and DAC B can be used for fine adjustment. The insertion of R1 in series with DAC B attenuates its contribution to the voltage sum node at the DAC C output.

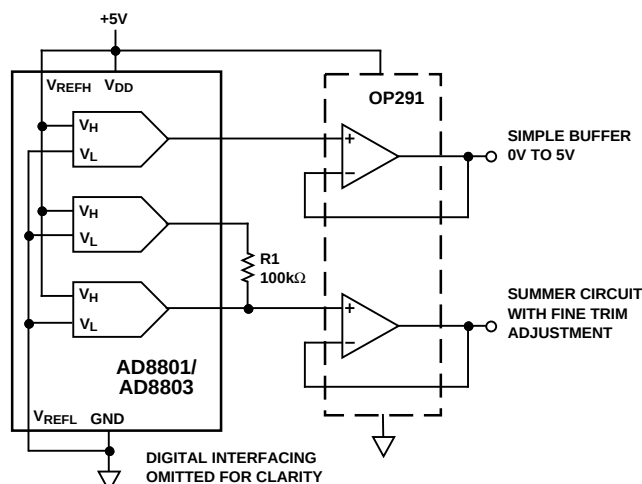


Figure 23. Buffering the AD8801/AD8803 Output

Increasing Output Voltage Swing

An external amplifier can also be used to extend the output voltage swing beyond the power supply rails of the AD8801/AD8803. This technique permits an easy digital interface for the DAC, while expanding the output swing to take advantage of higher voltage external power supplies. For example, DAC A of Figure 24 is configured to swing from -5 V to $+5\text{ V}$. The actual output voltage is given by:

$$V_{OUT} = \left(1 + \frac{R_E}{R_S}\right) \times \left(\frac{D}{256} \times 5\text{ V}\right) - 5\text{ V}$$

Where D is the DAC input value (i.e., 0 to 255). This circuit can be combined with the “fine/coarse” circuit of Figure 23 if, for example, a very accurate adjustment around 0 V is desired.

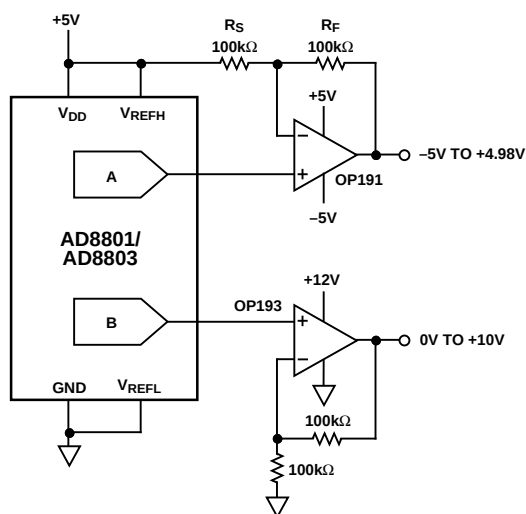


Figure 24. Increasing Output Voltage Swing

DAC B of Figure 24 is in a noninverting gain of two configuration, which increases the available output swing to $+10\text{ V}$. The feedback resistors can be adjusted to provide any scaling of the output voltage, within the limits of the external op amp power supplies.

Microcomputer Interfaces

The AD8801/AD8803 serial data input provides an easy interface to a variety of single-chip microcomputers (μCs). Many μCs have a built-in serial data capability that can be used for communicating with the DAC. In cases where no serial port is provided, or it is being used for some other purpose (such as an RS-232 communications interface), the AD8801/AD8803 can easily be addressed in software.

Eleven data bits are required to load a value into the AD8801/AD8803 (3 bits for the DAC address and 8 bits for the DAC value). If more than 11 bits are transmitted before the Chip Select input goes high, the extra (i.e., the most-significant) bits are ignored. This feature is valuable because most μCs only transmit data in 8-bit increments. Thus, the μC will send 16 bits to the DAC instead of 11 bits. The AD8801/AD8803 will only respond to the last 11 bits clocked into the SDI input, however, so the serial data interface is not affected.

An 8051 μC Interface

A typical interface between the AD8801/AD8803 and an 8051 μC is shown in Figure 25. This interface uses the 8051's internal serial port. The serial port is programmed for Mode 0 operation, which functions as a simple 8-bit shift register. The 8051's Port3.0 pin functions as the serial data output, while Port3.1 serves as the serial clock.

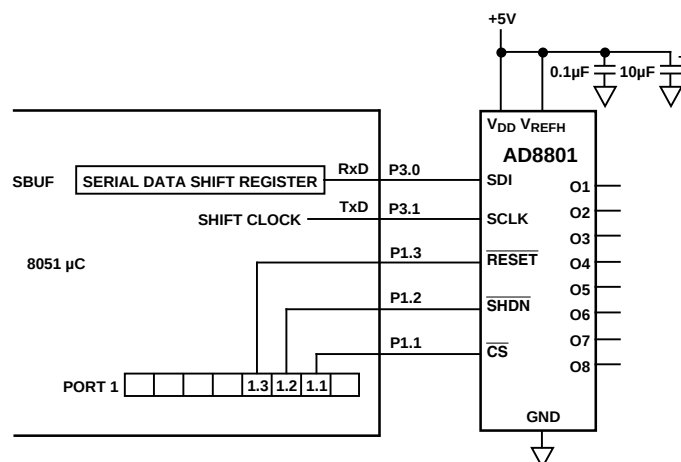


Figure 25. Interfacing the 8051 μC to an AD8801/AD8803, Using the Serial Port

When data is written to the Serial Buffer Register (SBUF, at Special Function Register location 99_H), the data is automatically converted to serial format and clocked out via Port3.0 and Port3.1. After 8 bits have been transmitted, the Transmit Interrupt flag (SCON.1) is set and the next 8 bits can be transmitted.

The AD8801 and AD8803 require the Chip Select to go low at the beginning of the serial data transfer. In addition, the SCLK input must be high when the Chip Select input goes high at the end of the transfer. The 8051's serial clock meets this requirement, since Port3.1 both begins and ends the serial data in the high state.

Software for the 8051 Interface

A software routine for the AD8801/AD8803 to 8051 interface is shown in Listing 1. The routine transfers the 8-bit data stored at data memory location DAC_VALUE to the AD8801/AD8803 DAC addressed by the contents of location DAC_ADDR.

AD8801/AD8803

```

;
; This subroutine loads an AD8801/AD8803 DAC from an 8051 microcomputer,
; using the 8051's serial port in MODE 0 (Shift Register Mode).
; The DAC value is stored at location DAC_VAL
; The DAC address is stored at location DAC_ADDR
;
; Variable declarations
;
PORT1          DATA          90H          ;SFR register for port 1
DAC_VALUE      DATA          40H          ;DAC Value
DAC_ADDR       DATA          41H          ;DAC Address
SHIFT1        DATA          042H         ;high byte of 16-bit answer
SHIFT2        DATA          043H         ;low byte of answer
SHIFT_COUNT    DATA          44H         ;
;
DO_8801:       ORG            100H         ;arbitrary start
               CLR            SCON.7       ;set serial
               CLR            SCON.6       ; data mode 0
               CLR            SCON.5
               CLR            SCON.1       ;clr transmit flag
               ORL            PORT1.1,#00001110B ;/RS, /SHDN, /CS high
               CLR            PORT1.1      ;set the /CS low
               MOV            SHIFT1,DAC_ADDR ;put DAC value in shift register
               ACALL          BYTESWAP     ;
               MOV            SBUF,SHIFT2  ;send the address byte
ADDR_WAIT:     JNB            SCON.1,ADDR_WAIT ;wait until 8 bits are sent
               CLR            SCON.1       ;clear the serial transmit flag
               MOV            SHIFT1,DAC_VALUE ;send the DAC value
               ACALL          BYTESWAP     ;
               MOV            SBUF,SHIFT2  ;
VALU_WAIT:     JNB            SCON.1,VALU_WAIT ;wait again
               CLR            SCON.1       ;clear serial flag
               SETB           PORT1.1      ;/CS high, latch data
               RET                  ; into AD8801
;
BYTESWAP:      MOV            SHIFT_COUNT,#8 ;Shift 8 bits
SWAP_LOOP:     MOV            A,SHIFT1     ;Get source byte
               RLC              A          ;Rotate MSB to carry
               MOV            SHIFT1,A     ;Save new source byte
               MOV            A,SHIFT2     ;Get destination byte
               RRC              A          ;Move carry to MSB
               MOV            SHIFT2,A     ;Save
               DJNZ           SHIFT_COUNT,SWAP_LOOP ;Done?
               RET
END

```

Listing 1. Software for the 8051 to AD8801/AD8803 Serial Port Interface

The subroutine begins by setting appropriate bits in the Serial Control register to configure the serial port for Mode 0 operation. Next the DAC's Chip Select input is set low to enable the AD8801/AD8803. The DAC address is obtained from memory location DAC_ADDR, adjusted to compensate for the 8051's serial data format, and moved to the serial buffer register. At this point, serial data transmission begins automatically. When all 8 bits have been sent, the Transmit Interrupt bit is set, and the subroutine then proceeds to send the DAC value stored at location DAC_VALUE. Finally the Chip Select input is returned high, causing the appropriate AD8801/AD8803 output voltage to change, and the subroutine ends.

The 8051 sends data out of its shift register LSB first, while the AD8801/AD8803 require data MSB first. The subroutine therefore includes a BYTESWAP subroutine to reformat the data. This routine transfers the MSB-first byte at location SHIFT1 to an LSB-first byte at location SHIFT2. The routine rotates the MSB of the first byte into the carry with a Rotate Left Carry instruction, then rotates the carry into the MSB of the second byte with a Rotate Right Carry instruction. After 8 loops, SHIFT2 contains the data in the proper format.

The BYTESWAP routine in Listing 1 is convenient because the DAC data can be calculated in normal LSB form. For example, producing a ramp voltage on a DAC is simply a matter of repeatedly incrementing the DAC_VALUE location and calling the LD_8801 subroutine.

If the μ C's hardware serial port is being used for other purposes, the AD8801/AD8803 can be loaded by using the parallel port. A typical parallel interface is shown in Figure 26. The serial data is transmitted to the DAC via the 8051's Port1.7 output, while Port1.6 acts as the serial clock.

Software for the interface of Figure 26 is contained in Listing 2. The subroutine will send the value stored at location DAC_VALUE to the AD8801/AD8803 DAC addressed by location DAC_ADDR. The program begins by setting the AD8801/AD8803's Serial Clock and Chip Select inputs high, then setting Chip Select low to start the serial interface process. The DAC address is loaded into the accumulator and three Rotate Right shifts are performed. This places the DAC address in the 3 MSBs of the accumulator. The address is then sent to the AD8801/AD8803 via the SEND_SERIAL subroutine. Next, the DAC value is loaded into the accumulator and sent to the AD8801/AD8803. Finally, the Chip Select input is set high to complete the data transfer.

```
; This 8051  $\mu$ C subroutine loads an AD8801 or AD8803 DAC with an 8-bit value,
; using the 8051's parallel port #1.
; The DAC value is stored at location DAC_VALUE
; The DAC address is stored at location DAC_ADDR
;
; Variable declarations
PORT1          DATA          90H                ;SFR register for port 1
DAC_VALUE      DATA          40H                ;DAC Value
DAC_ADDR       DATA          41H                ;DAC Address (0 through 7)
LOOPCOUNT     DATA          43H                ;COUNT LOOPS
;
LD_8803:        ORG            100H                ;arbitrary start
                ORL            PORT1,#11110000B    ;set CLK, /CS and /SHDN high,
                CLR            PORT1.5            ;Set Chip Select low
                MOV            LOOPCOUNT,#3        ;Address is 3 bits
                MOV            A,DAC_ADDR           ; Get DAC address
                RR              A                  ; Rotate the DAC
                RR              A                  ;address to the Most
                RR              A                  ;Significant Bits (MSBs)
                ACALL          SEND_SERIAL          ;Send the address
                MOV            LOOPCOUNT,#8        ;Do 8 bits of data
                MOV            A,DAC_VALUE
                ACALL          SEND_SERIAL          ;Send the data
                SETB           PORT1.5            ;Set /CS high
                RET                                ;DONE

SEND_SERIAL:    RLC              A                ;Move next bit to carry
                MOV            PORT1.7,C          ;Move data to SDI
                CLR            PORT1.6            ;Pulse the
                SETB           PORT1.6            ; CLK input
                DJNZ           LOOPCOUNT,SEND_SERIAL ;Loop if not done
                RET
END
```

Listing 2. Software for the 8051 to AD8801/AD8803 Parallel Port Interface

AD8801/AD8803

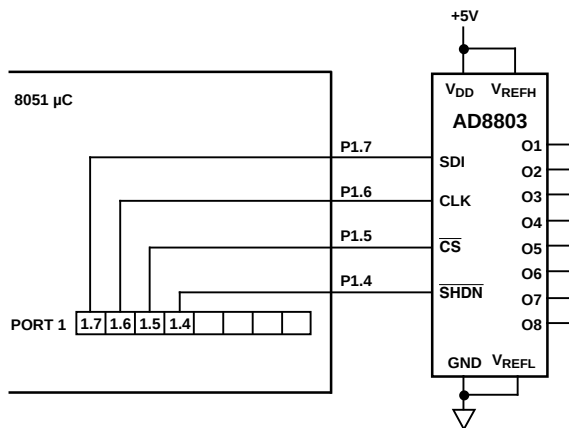


Figure 26. An AD8801/AD8803-8051 μ C Interface Using Parallel Port 1

Unlike the serial port interface of Figure 25, the parallel port interface only transmits 11 bits to the AD8801/AD8803. Also, the BYTESWAP subroutine is not required for the parallel interface, because data can be shifted out MSB first. However, the results of the two interface methods are exactly identical. In most cases, the decision on which method to use will be determined by whether or not the serial data port is available for communication with the AD8801/AD8803.

An MC68HC11-to-AD8801/AD8803 Interface

Like the 8051, the MC68HC11 includes a dedicated serial data port (labeled SPI). The SPI port provides an easy interface to the AD8801/AD8803 (Figure 27). The interface uses three lines of Port D for the serial data, and one or two lines from Port C to control the $\overline{\text{SHDN}}$ and $\overline{\text{RS}}$ (AD8801 only) inputs.

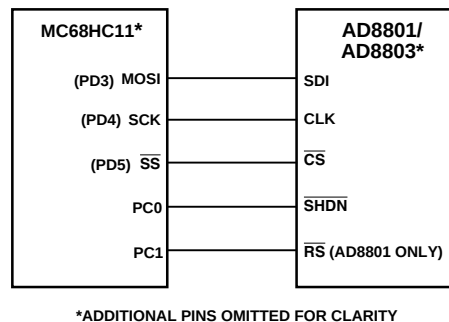


Figure 27. An AD8801/AD8803-to-MC68HC11 Interface

A software routine for loading the AD8801/AD8803 from a 68HC11 evaluation board is shown in Listing 3. First, the MC68HC11 is configured for SPI operation. Bits CPHA and CPOL define the SPI mode wherein the serial clock (SCK) is high at the beginning and end of transmission, and data is valid on the rising edge of SCK. This mode matches the requirements of the AD8801/AD8803. After the registers are saved on the stack, the DAC value and address are transferred to RAM and the AD8801/AD8803's $\overline{\text{CS}}$ is driven low. Next, the DAC's address byte is transferred to the SPDR register, which automatically initiates the SPI data transfer. The program tests the SPIF bit and loops until the data transfer is complete. Then the DAC value is sent to the SPI. When transmission of the second byte is complete, $\overline{\text{CS}}$ is driven high to load the new data and address into the AD8801/AD8803.

```

*
* AD8801/AD8803 to M68HC11 Interface Assembly Program
*
* M68HC11 Register definitions
*
PORTC      EQU          $1003      Port C control register
*                                     "0,0,0,0;0,0,RS/, SHDN/"
DDRC        EQU          $1007      Port C data direction
PORTD       EQU          $1008      Port D data register
*                                     "0,0,/CS,CLK;SDI,0,0,0"
DDRD        EQU          $1009      Port D data direction
SPCR        EQU          $1028      SPI control register
*                                     "SPIE,SPE,DWOM,MSTR;CPOL,CPHA,SPR1,SPR0"
SPSR        EQU          $1029      SPI status register
*                                     "SPIF,WCOL,0,MODE;0,0,0,0"
SPDR        EQU          $102A      SPI data register; Read-Buffer; Write-Shifter
*
* SDI RAM variables:
*                                     SDI1 is encoded from 0 (Hex) to 7 (Hex)
*                                     SDI2 is encoded from 00 (Hex) to FF (Hex)
*                                     AD8801/3 requires two 8-bit loads; upper 5 bits
*                                     of SDI1 are ignored. AD8801/3 address bits in last
*                                     three LSBs of SDI1.
SDI1        EQU          $00        SDI packed byte 1 "0,0,0,0;0,A2,A1,A0"
SDI2        EQU          $01        SDI packed byte 2 "DB7,DB6,DB5,DB4;DB3,DB2,DB1,DB0"
*
* Main Program
*
          ORG          $C000        Start of user's RAM in EVB
INIT        LDS          #$CFFF      Top of C page RAM
*
* Initialize Port C Outputs
*
          LDAA          #$03        0,0,0,0;0,0,1,1
*                                     /RS-Hi, /SHDN-Hi
          STAA          PORTC      Initialize Port C Outputs
          LDAA          #$03        0,0,0,0;0,0,1,1
          STAA          DDRC       /RS and /SHDN are now enabled as outputs
*
* Initialize Port D Outputs
*
          LDAA          #$20        0,0,1,0;0,0,0,0
*                                     /CS-Hi,/CLK-Lo,SDI-Lo
          STAA          PORTD      Initialize Port D Outputs
          LDAA          #$38        0,0,1,1;1,0,0,0
          STAA          DDRD       /CS,CLK, and SDI are now enabled as outputs
*
* Initialize SPI Interface
*
          LDAA          #$53
          STAA          SPCR        SPI is Master,CPHA=0,CPOL=0,Clk rate=E/32
*
* Call update subroutine
*
          BSR          UPDATE      Xfer 2 8-bit words to AD8402
          JMP          $E000       Restart BUFFALO
*
* Subroutine UPDATE
*
UPDATE      PSHX                Save registers X, Y, and A

```

AD8801/AD8803

```

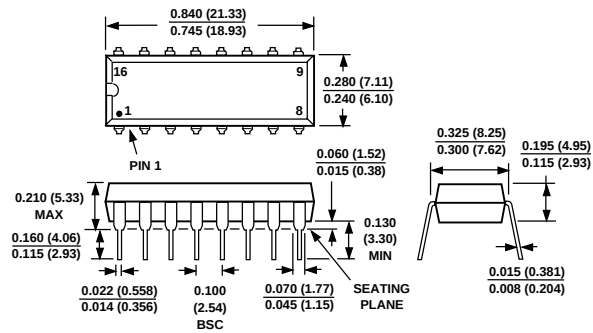
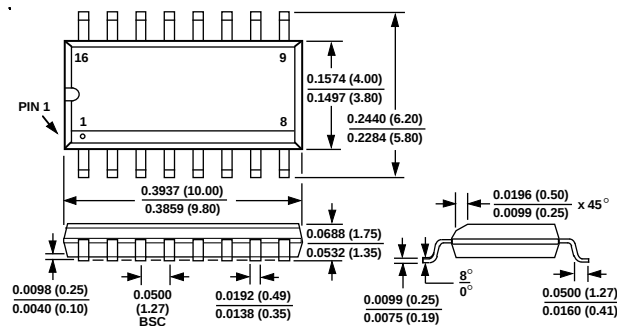
        PSHY
        PSHA
*
* Enter Contents of SDI1 Data Register
*
        LDAA      $0000      Hi-byte data loaded from memory
        STAA      SDI1       SDI1 = data in location 0000H
*
* Enter Contents of SDI2 Data Register
*
        LDAA      $0001      Low-byte data loaded from memory
        STAA      SDI2       SDI2 = Data in location 0001H
*
        LDX       #SDI1      Stack pointer at 1st byte to send via SDI
        LDY       #$1000     Stack pointer at on-chip registers
*
* Reset AD8801 to one-half scale (AD8803 does not have a Reset input)
*
        BCLR      PORTC,Y $02  Assert /RS
        BSET      PORTC,Y $02  De-assert /RS
*
* Get AD8801/03 ready for data input
*
        BCLR      PORTD,Y $20  Assert /CS
*
TFRLP   LDAA      0,X         Get a byte to transfer via SPI
        STAA      SPDR       Write SDI data reg to start xfer
*
WAIT    LDAA      SPSR       Loop to wait for SPIF
        BPL       WAIT       SPIF is the MSB of SPSR
                                (when SPIF is set, SPSR is negated)
*
        INX              Increment counter to next byte for xfer
        CPX          #SDI2+1 Are we done yet ?
        BNE         TFRLP    If not, xfer the second byte
*
* Update AD8801 output
*
        BSET      PORTD,Y $20  Latch register & update AD8801
*
        PULA              When done, restore registers X, Y & A
        PULY
        PULX
        RTS
** Return to Main Program **

```

Listing 3. AD8801/AD8803 to MC68HC11 Interface Program Source Code

OUTLINE DIMENSIONS

Dimensions shown in inches and (mm).

16-Pin Plastic DIP Package (N-16)**16-Pin Narrow Body SOIC Package (R-16A)**

Данный компонент на территории Российской Федерации

Вы можете приобрести в компании MosChip.

Для оперативного оформления запроса Вам необходимо перейти по данной ссылке:

<http://moschip.ru/get-element>

Вы можете разместить у нас заказ для любого Вашего проекта, будь то серийное производство или разработка единичного прибора.

В нашем ассортименте представлены ведущие мировые производители активных и пассивных электронных компонентов.

Нашей специализацией является поставка электронной компонентной базы двойного назначения, продукции таких производителей как XILINX, Intel (ex.ALTERA), Vicor, Microchip, Texas Instruments, Analog Devices, Mini-Circuits, Amphenol, Glenair.

Сотрудничество с глобальными дистрибьюторами электронных компонентов, предоставляет возможность заказывать и получать с международных складов практически любой перечень компонентов в оптимальные для Вас сроки.

На всех этапах разработки и производства наши партнеры могут получить квалифицированную поддержку опытных инженеров.

Система менеджмента качества компании отвечает требованиям в соответствии с ГОСТ Р ИСО 9001, ГОСТ РВ 0015-002 и ЭС РД 009

Офис по работе с юридическими лицами:

105318, г.Москва, ул.Щербаковская д.3, офис 1107, 1118, ДЦ «Щербаковский»

Телефон: +7 495 668-12-70 (многоканальный)

Факс: +7 495 668-12-70 (доб.304)

E-mail: info@moschip.ru

Skype отдела продаж:

moschip.ru

moschip.ru_4

moschip.ru_6

moschip.ru_9