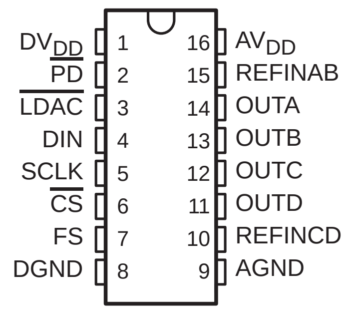
- **Four 12-Bit D/A Converters**
- **Programmable Settling Time of Either 3 µs or 9 µs Typ**
- **TMS320, (Q)SPI, and Microwire Compatible Serial Interface**
- **Internal Power-On Reset**
- **Low Power Consumption:**
  **8 mW, Slow Mode – 5-V Supply**
  **3.6 mW, Slow Mode – 3-V Supply**
- **Reference Input Buffer**
- **Voltage Output Range . . . 2× the Reference Input Voltage**
- **Monotonic Over Temperature**
- **Dual 2.7-V to 5.5-V Supply (Separate Digital and Analog Supplies)**
- **Hardware Power Down (10 nA)**
- **Software Power Down (10 nA)**
- **Simultaneous Update**

## applications

- **Battery Powered Test Instruments**
- **Digital Offset and Gain Adjustment**
- **Industrial Process Controls**
- **Machine and Motion Control Devices**
- **Communications**
- **Arbitrary Waveform Generation**

## description

**D OR PW PACKAGE**
**(TOP VIEW)**




The TLV5614 is a quadruple 12-bit voltage output digital-to-analog converter (DAC) with a flexible 4-wire serial interface. The 4-wire serial interface allows glueless interface to TMS320, SPI, QSPI, and Microwire serial ports. The TLV5614 is programmed with a 16-bit serial word comprised of a DAC address, individual DAC control bits, and a 12-bit DAC value. The device has provision for two supplies: one digital supply for the serial interface (via pins $DV_{DD}$ and DGND), and one for the DACs, reference buffers, and output buffers (via pins $AV_{DD}$ and AGND). Each supply is independent of the other, and can be any value between 2.7 V and 5.5 V. The dual supplies allow a typical application where the DAC will be controlled via a microprocessor operating on a 3 V supply (also used on pins $DV_{DD}$ and DGND), with the DACs operating on a 5 V supply. Of course, the digital and anlog supplies can be tied together.

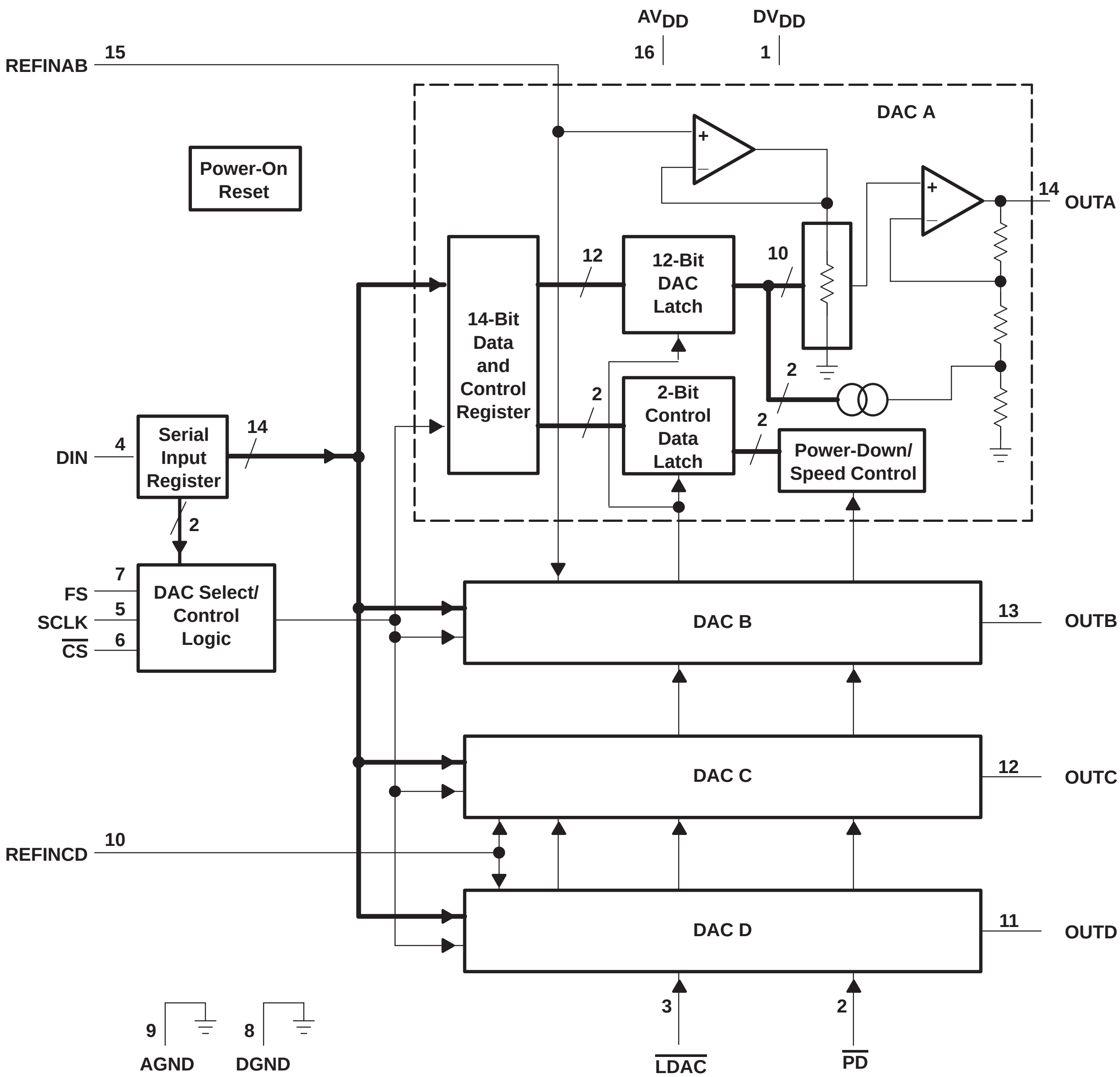
The resistor string output voltage is buffered by a x2 gain rail-to-rail output buffer. The buffer features a Class AB output stage to improve stability and reduce settling time. A rail-to-rail output stage and a power-down mode makes it ideal for single voltage, battery based applications. The settling time of the DAC is programmable to allow the designer to optimize speed versus power dissipation. The settling time is chosen by the control bits within the 16-bit serial input string. A high-impedance buffer is integrated on the REFINAB and REFINCD terminals to reduce the need for a low source impedance drive to the terminal. REFINAB and REFINCD allow DACs A and B to have a different reference voltage then DACs C and D.

The TLC5614 is implemented with a CMOS process and is available in a 16-terminal SOIC package. The TLV5614C is characterized for operation from 0°C to 70°C. The TLV5614I is characterized for operation from –40°C to 85°C.

**AVAILABLE OPTIONS**

| $T_A$ | PACKAGE | |
|---|---|---|
| | SOIC (D) | TSSOP (PW) |
| 0°C to 70°C | TLV5614CD | TLV5614CPW |
| –40°C to 85°C | TLV5614ID | TLV5614IPW |

## functional block diagram



## Terminal Functions

| TERMINAL NAME | TERMINAL NO. | I/O | DESCRIPTION |
|---|---|---|---|
| AGND | 9 | | Analog ground |
| $AV_{DD}$ | 16 | | Analog supply |
| $\overline{CS}$ | 6 | I | Chip select. This terminal is active low. |
| DGND | 8 | | Digital ground |
| DIN | 4 | I | Serial data input |
| $DV_{DD}$ | 1 | | Digital supply |
| FS | 7 | I | Frame sync input. The falling edge of the frame sync pulse indicates the start of a serial data frame shifted out to the TLV5614. |
| $\overline{PD}$ | 2 | I | Power down pin. Powers down all DACs (overriding their individual power down settings), and all output stages. This terminal is active low. |
| $\overline{LDAC}$ | 3 | I | Load DAC. When the $\overline{LDAC}$ signal is high, no DAC output updates occur when the input digital data is read into the serial interface. The DAC outputs are only updated when $\overline{LDAC}$ is low. |
| REFINAB | 15 | I | Voltage reference input for DACs A and B. |
| REFINCD | 10 | I | Voltage reference input for DACs C and D. |
| SCLK | 5 | I | Serial Clock input |
| OUTA | 14 | O | DACA output |
| OUTB | 13 | O | DACB output |
| OUTC | 12 | O | DACC output |
| OUTD | 11 | O | DACD output |

### absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage, ($DV_{DD}$, $AV_{DD}$ to GND) . . . 7 V
Supply voltage difference, ($AV_{DD}$ to $DV_{DD}$) . . . –2.8 V to 2.8 V
Digital input voltage range . . . –0.3 V to $DV_{DD}$ + 0.3 V
Reference input voltage range . . . –0.3 V to $AV_{DD}$ + 0.3 V
Operating free-air temperature range, $T_A$: TLV5614C . . . 0°C to 70°C
TLV5614I . . . –40°C to 85°C
Storage temperature range, $T_{stg}$ . . . –65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds . . . 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 under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

## recommended operating conditions

| | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|
| Supply voltage, $AV_{DD}$, $DV_{DD}$ | 5-V supply | 4.5 | 5 | 5.5 | V |
| | 3-V supply | 2.7 | 3 | 3.3 | |
| High-level digital input, $V_{IH}$ | $DV_{DD}$ = 2.7 V to 5.5 V | 2 | | | V |
| Low-level digital input, $V_{IL}$ | $DV_{DD}$ = 2.7 V to 5.5 V | | | 0.8 | V |
| Reference voltage, $V_{ref}$ to REFINAB, REFINCD terminal | 5-V supply, See Note 1 | 0 | 2.048 | $V_{DD}$–1.5 | V |
| | 3-V supply, See Note 1 | 0 | 1.024 | $V_{DD}$–1.5 | |
| Load resistance, $R_L$ | | 2 | 10 | | kΩ |
| Load capacitance, $C_L$ | | | | 100 | pF |
| Serial clock rate, SCLK | | | | 20 | MHz |
| Operating free-air temperature | TLV5614C | 0 | | 70 | °C |
| | TLV5614I | –40 | | 85 | |

NOTE 1: Voltages greater than $AV_{DD}/2$ will cause output saturation for large DAC codes.

## electrical characteristics over recommended operating free-air temperature range, supply voltages, and reference voltages (unless otherwise noted)

### static DAC specifications

| | PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| | Resolution | | | 12 | | | bits |
| | Integral nonlinearity (INL), end point adjusted | | See Note 2 | | ±1.5 | ±4 | LSB |
| | Differential nonlinearity (DNL) | | See Note 3 | | ±0.5 | ±1 | LSB |
| $E_{ZS}$ | Zero scale error (offset error at zero scale) | | See Note 4 | | | ±12 | mV |
| | Zero scale error temperature coefficient | | See Note 5 | | 10 | | ppm/°C |
| $E_G$ | Gain error | | See Note 6 | | | ±0.6 | % of FS voltage |
| | Gain error temperature coefficient | | See Note 7 | | 10 | | ppm/°C |
| PSRR | Power supply rejection ratio | Zero scale | See Notes 8 and 9 | | –80 | | dB |
| | | Full scale | | | –80 | | dB |

NOTES: 2. The relative accuracy or integral nonlinearity (INL) sometimes referred to as linearity error, is the maximum deviation of the output from the line between zero and full scale excluding the effects of zero code and full-scale errors.
3. The differential nonlinearity (DNL) sometimes referred to as differential error, is the difference between the measured and ideal 1 LSB amplitude change of any two adjacent codes. Monotonic means the output voltage changes in the same direction (or remains constant) as a change in the digital input code.
4. Zero-scale error is the deviation from zero voltage output when the digital input code is zero.
5. Zero-scale-error temperature coefficient is given by: $E_{ZS}\ TC = [E_{ZS}(T_{max}) - E_{ZS}(T_{min})]/V_{ref} \times 10^6/(T_{max} - T_{min})$.
6. Gain error is the deviation from the ideal output ($2\ V_{ref}$ – 1 LSB) with an output load of 10 kΩ excluding the effects of the zero-error.
7. Gain temperature coefficient is given by: $E_G\ TC = [E_G(T_{max}) - E_G(T_{min})]/V_{ref} \times 10^6/(T_{max} - T_{min})$.
8. Zero-scale-error rejection ratio (EZS–RR) is measured by varying the $AV_{DD}$ from 5 ± 0.5 V and 3 ± 0.5 V dc, and measuring the proportion of this signal imposed on the zero-code output voltage.
9. Full-scale rejection ratio (EG-RR) is measured by varying the $AV_{DD}$ from 5 ± 0.5 V and 3 ± 0.5 V dc and measuring the proportion of this signal imposed on the full-scale output voltage after subtracting the zero scale change.

## electrical characteristics over recommended operating free-air temperature range, supply voltages, and reference voltages (unless otherwise noted) (continued)

### individual DAC output specifications

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $V_O$ | Voltage output range | $R_L$ = 10 kΩ | 0 | | $AV_{DD}$–0.4 | V |
| | Output load regulation accuracy | $R_L$ = 2 kΩ vs 10 kΩ | | 0.1 | 0.25 | % of FS voltage |

### reference inputs (REFINAB, REFINCD)

| PARAMETER | | TEST CONDITIONS | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| $V_I$ | Input voltage range | See Note 10 | | 0 | | $AV_{DD}$–1.5 | V |
| $R_I$ | Input resistance | | | | 10 | | MΩ |
| $C_I$ | Input capacitance | | | | 5 | | pF |
| | Reference feed through | REFIN = 1 $V_{pp}$ at 1 kHz + 1.024 V dc (see Note 11) | | | –75 | | dB |
| | Reference input bandwidth | REFIN = 0.2 $V_{pp}$ + 1.024 V dc large signal | Slow | | 0.5 | | MHz |
| | | | Fast | | 1 | | |

NOTES: 10. Reference input voltages greater than $V_{DD}$/2 will cause output saturation for large DAC codes.
11. Reference feedthrough is measured at the DAC output with an input code = 000 hex and a $V_{ref}$ (REFINAB or REFINCD) input = 1.024 Vdc + 1 $V_{pp}$ at 1 kHz.

### digital inputs (DIN, $\overline{CS}$, $\overline{LDAC}$, $\overline{PD}$)

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $I_{IH}$ | High-level digital input current | $V_I = V_{DD}$ | | | ±1 | μA |
| $I_{IL}$ | Low-level digital input current | $V_I$ = 0 V | | | ±1 | μA |
| $C_I$ | Input capacitance | | | 3 | | pF |

### power supply

| PARAMETER | | TEST CONDITIONS | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| $I_{DD}$ | Power supply current | 5-V supply, No load, Clock running, All inputs 0 V or $V_{DD}$ | Slow | | 1.6 | 2.4 | mA |
| | | | Fast | | 3.8 | 5.6 | |
| | | 3-V supply, No load, Clock running, All inputs 0 V or $DV_{DD}$ | Slow | | 1.2 | 1.8 | mA |
| | | | Fast | | 3.2 | 4.8 | |
| | Power down supply current (see Figure 12) | | | | 10 | | nA |

## electrical characteristics over recommended operating free-air temperature range, supply voltages, and reference voltages (unless otherwise noted) (continued)

### analog output dynamic performance

| PARAMETER | | TEST CONDITIONS | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| SR | Output slew rate | $C_L$ = 100 pF, $R_L$ = 10 kΩ, $V_O$ = 10% to 90%, $V_{ref}$ = 2.048 V, 1024 V | Fast | | 5 | | V/μs |
| | | | Slow | | 1 | | V/μs |
| $t_s$ | Output settling time | To ± 0.5 LSB, $C_L$ = 100 pF, $R_L$ = 10 kΩ, See Notes 12 and 14 | Fast | | 3 | 5.5 | μs |
| | | | Slow | | 9 | 20 | |
| $t_{s(c)}$ | Output settling time, code to code | To ± 0.5 LSB, $C_L$ = 100 pF, $R_L$ = 10 kΩ, See Note 15 | Fast | | 1 | | μs |
| | | | Slow | | 2 | | |
| | Glitch energy | Code transition from 7FF to 800 | | | 10 | | nV-sec |
| SNR | Signal-to-noise ratio | Sinewave generated by DAC, Reference voltage = 1.024 at 3 V and 2.048 at 5 V, $f_s$ = 400 KSPS, $f_{OUT}$ = 1.1 kHz sinewave, $C_L$ = 100 pF, $R_L$ = 10 kΩ, BW = 20 kHz | | | 74 | | dB |
| S/(N+D) | Signal to noise + distortion | | | | 66 | | |
| THD | Total harmonic Distortion | | | | –68 | | |
| SFDR | Spurious free dynamic range | | | | 70 | | |

NOTES: 12. Settling time is the time for the output signal to remain within ±0.5 LSB of the final measured value for a digital input code change ofFFF hex to 080 hex for 080 hex to FFF hex.
13. Settling time is the time for the output signal to remain within ±0.5 LSB of the final measured value for a digital input code change of one count.
14. Limits are ensured by design and characterization, but are not production tested.

## electrical characteristics over recommended operating free-air temperature range, supply voltages, and reference voltages (unless otherwise noted) (continued)

### digital input timing requirements

| | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|
| $t_{su(CS–FS)}$ | Setup time, $\overline{CS}$ low before FS↓ | 10 | | | ns |
| $t_{su(FS–CK)}$ | Setup time, FS low before first negative SCLK edge | 8 | | | ns |
| $t_{su(C16–FS)}$ | Setup time, sixteenth negative edge after FS low on which bit D0 is sampled before rising edge of FS | 10 | | | ns |
| $t_{su(C16–CS)}$ | Setup time, sixteenth positive SCLK edge (first positive after D0 is sampled) before $\overline{CS}$ rising edge. If FS is used instead of the sixteenth positive edge to update the DAC, then the setup time is between the FS rising edge and $\overline{CS}$ rising edge. | 10 | | | ns |
| $t_{wH}$ | Pulse duration, SCLK high | 25 | | | ns |
| $t_{wL}$ | Pulse duration, SCLK low | 25 | | | ns |
| $t_{su(D)}$ | Setup time, data ready before SCLK falling edge | 8 | | | ns |
| $t_{h(D)}$ | Hold time, data held valid after SCLK falling edge | 5 | | | ns |
| $t_{wH(FS)}$ | Pulse duration, FS high | 20 | | | ns |

## PARAMETER MEASUREMENT INFORMATION

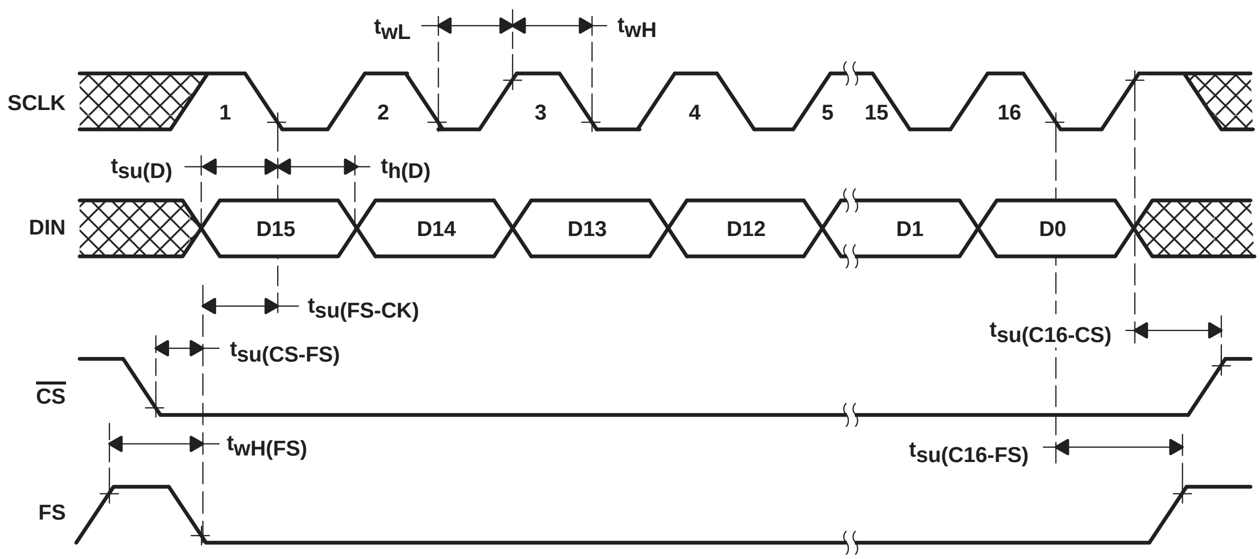



**Figure 1. Timing Diagram**

## TYPICAL CHARACTERISTICS

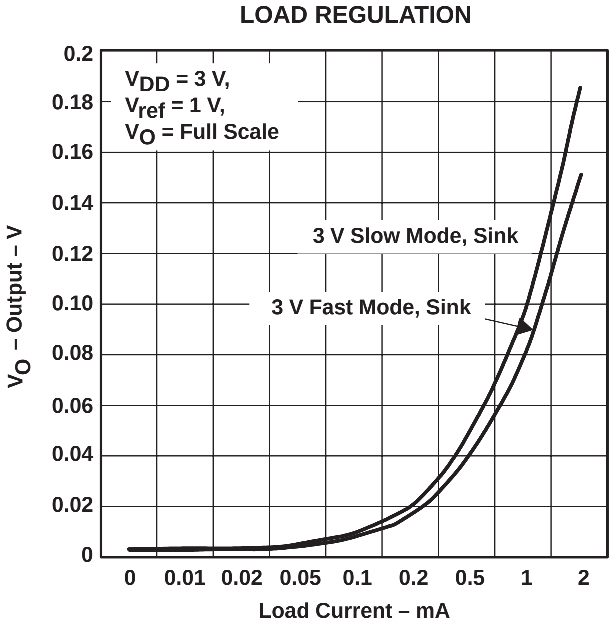



Figure 2

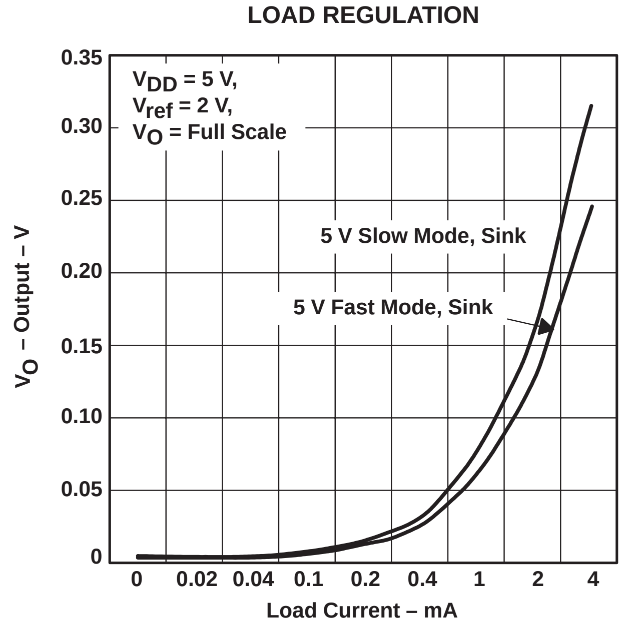



Figure 3

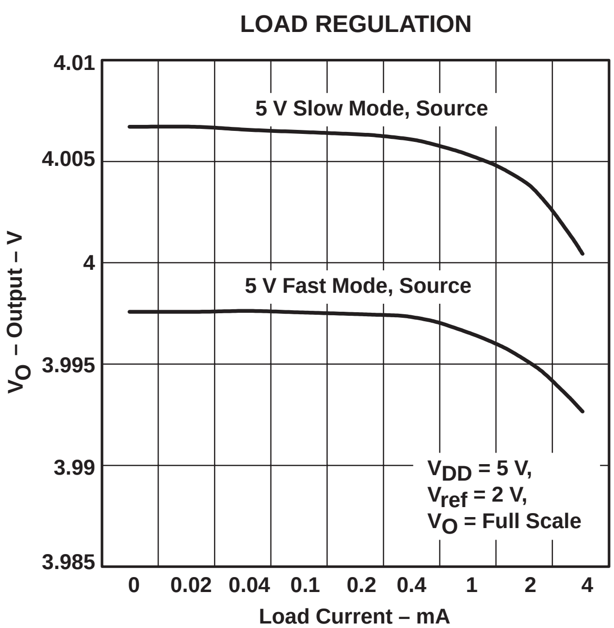



Figure 4

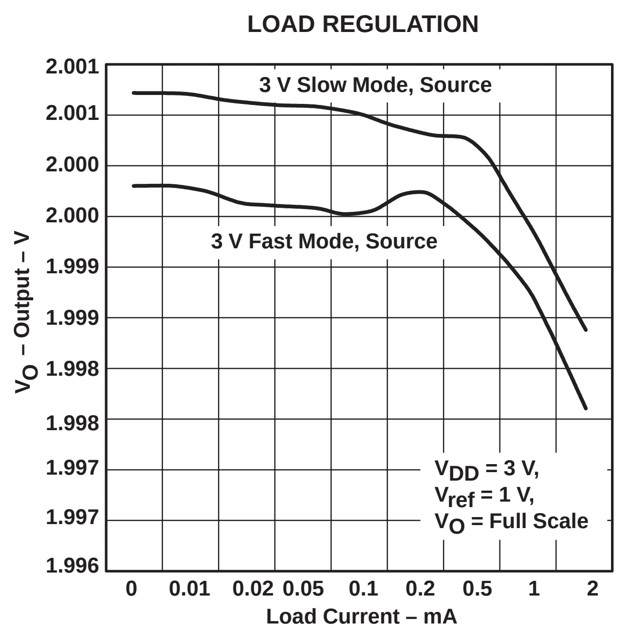



Figure 5

## TYPICAL CHARACTERISTICS

**SUPPLY CURRENT vs TEMPERATURE**

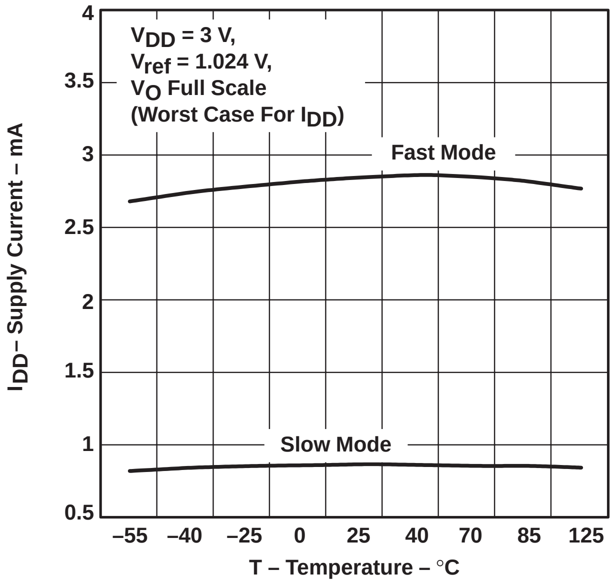



Figure 6

**SUPPLY CURRENT vs TEMPERATURE**

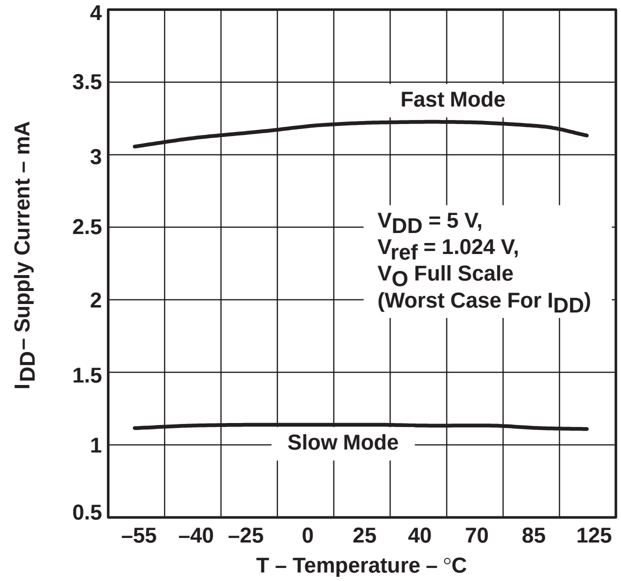



Figure 7

**TOTAL HARMONIC DISTORTION vs FREQUENCY**

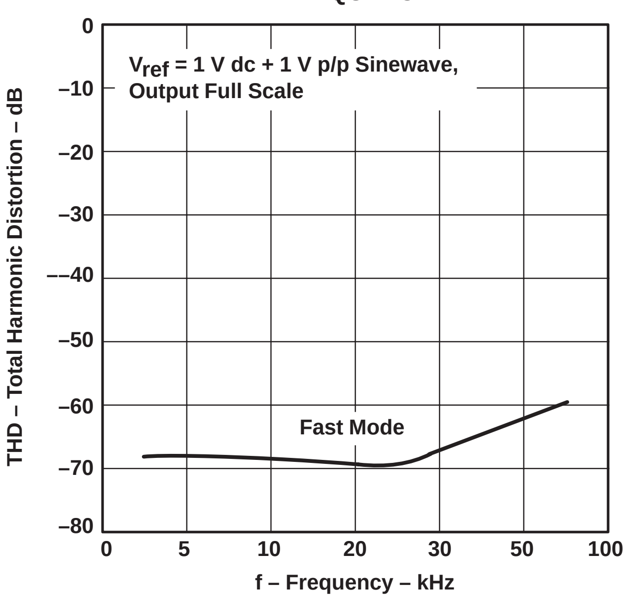



Figure 8

**TOTAL HARMONIC DISTORTION vs FREQUENCY**

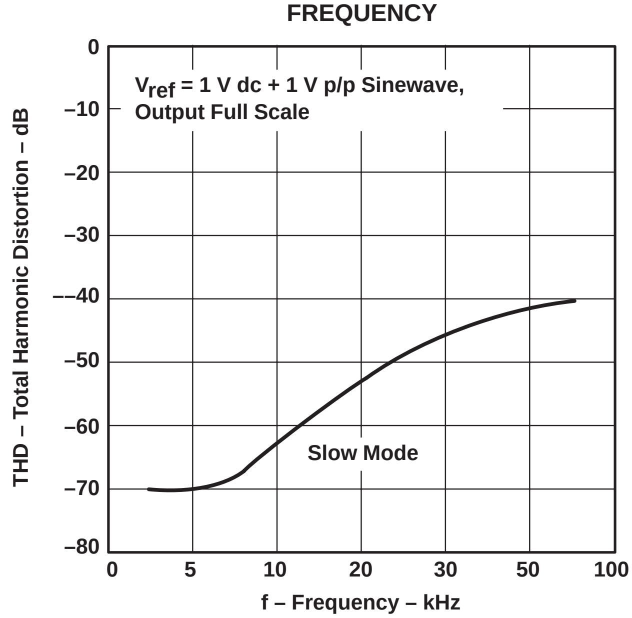



Figure 9

## TYPICAL CHARACTERISTICS

**TOTAL HARMONIC DISTORTION AND NOISE**
**vs**
**FREQUENCY**

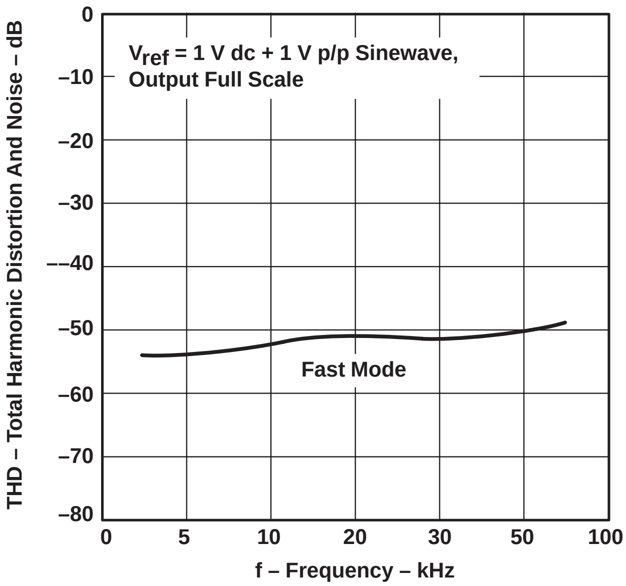



**Figure 10**

**TOTAL HARMONIC DISTORTION AND NOISE**
**vs**
**FREQUENCY**

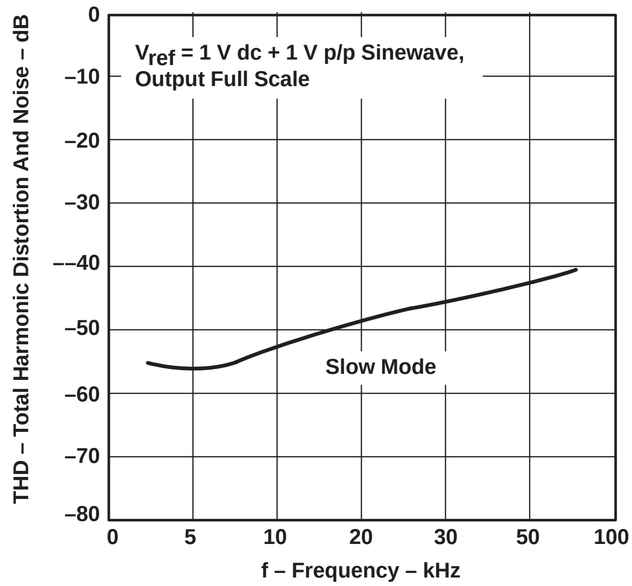



**Figure 11**

**SUPPLY CURRENT**
**vs**
**TIME**
**(WHEN ENTERING POWER-DOWN MODE)**

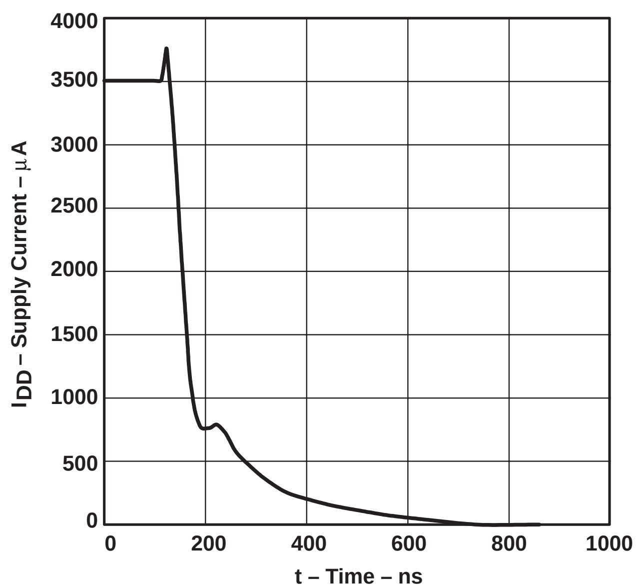



**Figure 12**

## TYPICAL CHARACTERISTICS

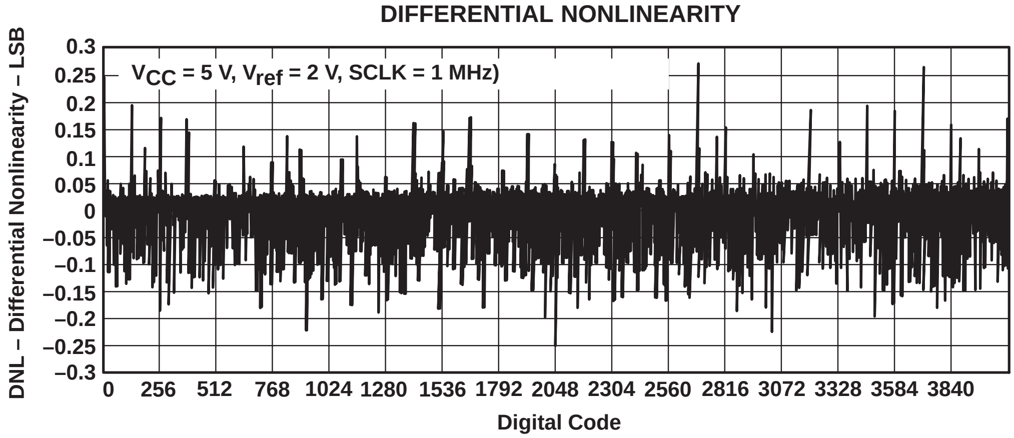



**Figure 13**

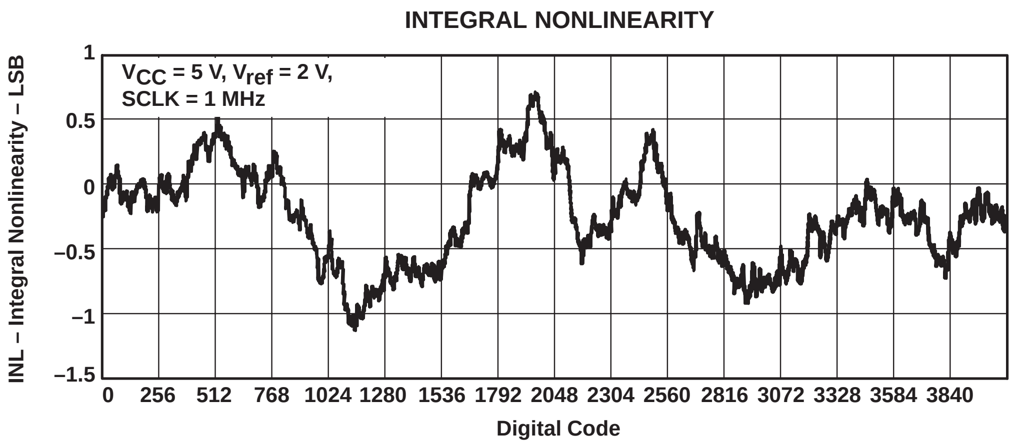



**Figure 14**

## APPLICATION INFORMATION

### general function

The TLV5614 is a 12-bit single supply DAC based on a resistor string architecture. The device consists of a serial interface, speed and power down control logic, a reference input buffer, a resistor string, and a rail-to-rail output buffer.

The output voltage (full scale determined by external reference) is given by:

$$2\ \mathrm{REF}\ \frac{\mathrm{CODE}}{0\mathrm{x}1000}\ [\mathrm{V}]$$

Where REF is the reference voltage and CODE is the digital input value within the range of 0x000 to 0xFFF. A power-on reset initially resets the internal latches to a defined state (all bits zero).

### serial interface

Explanation of data transfer: First, the device has to be enabled with $\overline{\mathrm{CS}}$ set to low. Then, a falling edge of FS starts shifting the data bit-per-bit (starting with the MSB) to the internal register on the falling edges of SCLK. After 16 bits have been transferred or FS rises, the content of the shift register is moved to the DAC latch which updates the voltage output to the new level.

The serial interface of the TLV5614 can be used in two basic modes:

- four wire (with chip select)
- three wire (without chip select)

Using chip select (four wire mode), it is possible to have more than one device connected to the serial port of the data source (DSP or microcontroller). The interface is compatible with the TMS320 family. Figure 15 shows an example with two TLV5614s connected directly to a TMS320 DSP.

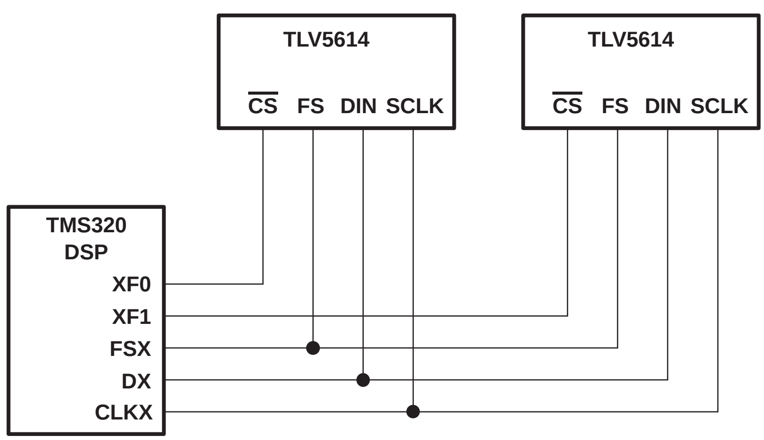



**Figure 15. TMS320 Interface**

## APPLICATION INFORMATION

### serial interface (continued)

If there is no need to have more than one device on the serial bus, then $\overline{CS}$ can be tied low. Figure 16 shows an example of how to connect the TLV5614 to a TMS320, SPI, or Microwire port using only three pins.

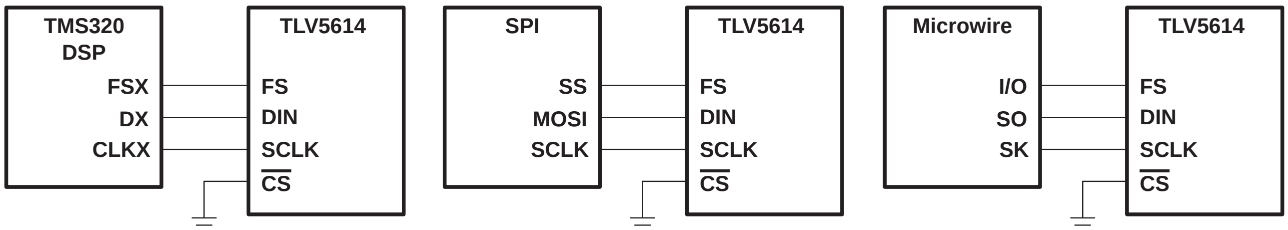



**Figure 16. Three-Wire Interface**

Notes on SPI and Microwire: Before the controller starts the data transfer, the software has to generate a falling edge on the I/O pin connected to FS. If the word width is 8 bits (SPI and Microwire), two write operations must be performed to program the TLV5614. After the write operation(s), the DAC output is updated automatically on the sixteenth positive clock edge.

### serial clock frequency and update rate

The maximum serial clock frequency is given by:

$$f_{SCLKmax} = \frac{1}{t_{wH(min)} + t_{wL(min)}} = 20\ \text{MHz}$$

The maximum update rate is:

$$f_{UPDATEmax} = \frac{1}{16\left(t_{wH(min)} + t_{wL(min)}\right)} = 1.25\ \text{MHz}$$

Note that the maximum update rate is a theoretical value for the serial interface since the settling time of the TLV5614 has to be considered also.

### data format

The 16-bit data word for the TLV5614 consists of two parts:

- Control bits (D15 . . . D12)
- New DAC value (D11 . . . D0)

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A1 | A0 | PWR | SPD | New DAC value (12 bits) | | | | | | | | | | | |

X: don’t care
SPD: Speed control bit. 1 → fast mode 0 → slow mode
PWR: Power control bit. 1 → power down 0 → normal operation

## APPLICATION INFORMATION

In power down mode, all amplifiers within the TLV5614 are disabled. A particular DAC (A, B, C, D) of the TLV5614 is selected by A1 and A0 within the input word.

| A1 | A0 | DAC |
|---|---|---|
| 0 | 0 | A |
| 0 | 1 | B |
| 1 | 0 | C |
| 1 | 1 | D |

## TLV5614 interfaced to TMS320C203 DSP

### hardware interfacing

Figure 17 shows an example of how to connect the TLV5614 to a TMS320C203 DSP. The serial port is configured in burst mode, with FSX generated by the TMS320C203 to provide the frame sync (FS) input to the TLV5614. Data is transmitted on the DX line, with the serial clock input on the CLKX line. The general-purpose input/output port bits IO0 and IO1 are used to generate the chip select ($\overline{CS}$) and DAC latch update ($\overline{LDAC}$) inputs to the TLV5614. The active low power down ($\overline{PD}$) is pulled high all the time to ensure the DACs are enabled.

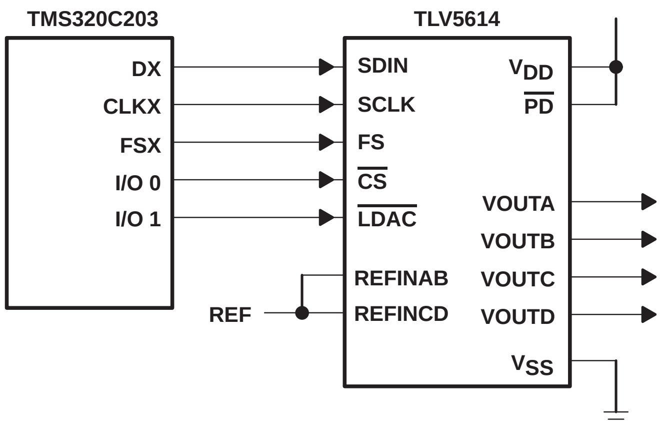



**Figure 17. TLV5614 Interfaced with TMS320C203**

### software

The application example outputs a differential in-phase (sine) signal between the VOUTA and VOUTB pins, and it’s quadrature (cosine) signal as the differential signal between VOUTC and VOUTD.

The on-chip timer is used to generate interrupts at a fixed frequency. The related interrupt service routine pulses $\overline{LDAC}$ low to update all 4 DACs simultaneously, then fetches and writes the next sample to all 4 DACs. The samples are stored in a look-up table, which describes two full periods of a sine wave.

The synchronous serial port of the DSP is used in burst mode. In this mode, the processor generates an FS pulse preceding the MSB of every data word. If multiple, contiguous words are transmitted, a violation of the tsu(C16–FS) timing requirement will occur. To avoid this, the program waits until the transmission of the previous word has been completed.

## APPLICATION INFORMATION

```
;-------------------------------------------------------------------------------------
; Processor:  TMS320C203 runnning at 40 MHz
;
; Description:
;
; This program generates a differential in–phase (sine) on (OUTA–OUTB) and it’s
; quadrature (cosine) as a differential signal on (OUTC–OUTD).
;
; The DAC codes for the signal samples are stored as a table of 64 12–bit values,
; describing 2 periods of a sine function. A rolling pointer is used to address the
; table location in the first period of this waveform, from which the DAC A samples
; are read. The samples for the other 3 DACs are read at an offset to this rolling
; pointer:
;  DAC    Function      Offset from rolling pointer
;   A     sine          0
;   B     inverse sine  16
;   C     cosine        8
;   D     inverse cosine24
;
; The on–chip timer is used to generate interrupts at a fixed rate. The interrupt
; service routine first pulses LDAC low to update all DACs simultaneously
; with the values which were written to them in the previous interrupt. Then all
; 4 DAC values are fetched and written out through the synchronous serial interface
; Finally, the rolling pointer is incremented to address the next sample, ready for
; the next interrupt.
;
; © 1998, Texas Instruments Inc.
;------------------------------------------------------------------------------------------
;------------------------------ I/O and memory mapped regs -----------------------------
       .include ”regs.asm”
;-------jump vectors -------------------------------------------------------------------
       .ps    0h
       b      start
       b      int1
       b      int23
       b      timer_isr;
----------- variables ---------------------------------------------------------------
temp          .equ   0060h
r_ptr         .equ   0061h
iosr_stat     .equ   0062h
DACa_ptr      .equ   0063h
DACb_ptr      .equ   0064h
DACc_ptr      .equ   0065h
DACd_ptr      .equ   0066h
;-----------constants------------------------------------------------------------------
; DAC control bits to be OR’ed onto data
; all fast mode
DACa_control .equ   01000h
DACb_control .equ   05000h
DACc_control .equ   09000h
DACd_control .equ   0d000h
;----------- tables --------------------------------
    .ds    02000h
sinevals
    .word  00800h
    .word  0097Ch
    .word  00AE9h
    .word  00C3Ah
    .word  00D61h
    .word  00E53h
    .word  00F07h
    .word  00F76h
    .word  00F9Ch
    .word  00F76h
    .word  00F07h
    .word  00E53h
```

## APPLICATION INFORMATION

```
.word  00D61h
.word  00C3Ah
.word  00AE9h
.word  0097Ch
.word  00800h
.word  00684h
.word  00517h
.word  003C6h
.word  0029Fh
.word  001ADh
.word  000F9h
.word  0008Ah
.word  00064h
.word  0008Ah
.word  000F9h
.word  001ADh
.word  0029Fh
.word  003C6h
.word  00517h
.word  00684h
.word  00800h
.word  0097Ch
.word  00AE9h
.word  00C3Ah
.word  00D61h
.word  00E53h
.word  00F07h
.word  00F76h
.word  00F9Ch
.word  00F76h
.word  00F07h
.word  00E53h
.word  00D61h
.word  00C3Ah
.word  00AE9h
.word  0097Ch
.word  00800h
.word  00684h
.word  00517h
.word  003C6h
.word  0029Fh
.word  001ADh
.word  000F9h
.word  0008Ah
.word  00064h
.word  0008Ah
.word  000F9h
.word  001ADh
.word  0029Fh
.word  003C6h
.word  00517h
.word  00684h
```

## APPLICATION INFORMATION

```
;-------------------------------------------------------------------------------------
; Main Program
;-------------------------------------------------------------------------------------
      .ps    1000h
      .entry
start
;-------------------------------------------------------------------------------------
; disable interrupts
;-------------------------------------------------------------------------------------
      setc   INTM      ; disable maskable interrupts
      splk   #0ffffh, IFR; clear all interrupts
      splk   #0004h, IMR; timer interrupts unmasked
;-------------------------------------------------------------------------------------
; set up the timer
; timer period set by values in PRD and TDDR
; period = (CLKOUT1 period) x (1+PRD) x (1+TDDR)
; examples for TMS320C203 with 40MHz main clock
; Timer rate     TDDR   PRD
;  80 kHz         9     24 (18h)
;  50 kHz         9     39 (27h)
;-------------------------------------------------------------------------------------
prd_val.equ      0018h
tcr_val.equ      0029h
      splk   #0000h, temp; clear timer
      out    temp, TIM
      splk   #prd_val, temp; set PRD
      out    temp, PRD
      splk   #tcr_val, temp; set TDDR, and TRB=1 for auto-reload
      out    temp, TCR
;-------------------------------------------------------------------------------------
; Configure IO0/1 as outputs to be :
; IO0  CS  - and set high
; IO1  LDAC   - and set high
;-------------------------------------------------------------------------------------
      in     temp, ASPCR; configure as output
      lacl   temp
      or     #0003h
      sacl   temp
      out    temp, ASPCR
      in     temp, IOSR; set them high
      lacl   temp
      or     #0003h
      sacl   temp
      out    temp, IOSR
;-------------------------------------------------------------------------------------
; set up serial port for
; SSPCR.TXM=1    Transmit mode - generate FSX
; SSPCR.MCM=1    Clock mode - internal clock source
; SSPCR.FSM=1    Burst mode
;-------------------------------------------------------------------------------------
      splk   #0000Eh, temp
      out    temp, SSPCR; reset transmitter
      splk   #0002Eh, temp
      out    temp,SSPCR
;-------------------------------------------------------------------------------------
; reset the rolling pointer
;-------------------------------------------------------------------------------------
      lacl   #000h
      sacl   r_ptr
;-------------------------------------------------------------------------------------
; enable interrupts
;-------------------------------------------------------------------------------------
      clrc   INTM      ; enable maskable interrupts
;-------------------------------------------------------------------------------------
; loop forever!
;-------------------------------------------------------------------------------------
```

## APPLICATION INFORMATION

```
next   idle             ;wait for interrupt
       b      next
;------------------------------------------------------------------------------------------
; all else fails stop here
;------------------------------------------------------------------------------------------
done   b      done      ;hang there
;------------------------------------------------------------------------------------------
; Interrupt Service Routines
;------------------------------------------------------------------------------------------
int1   ret       ; do nothing and return
int23  ret       ; do nothing and return
timer_isr:
    in     iosr_stat, IOSR; store IOSR value into variable space
    lacl   iosr_stat     ; load acc with iosr status
    and    #0FFFDh       ; reset IO1 – LDAC low
    sacl   temp          ;
    out    temp, IOSR    ;
    or     #0002h        ; set IO1 – LDAC high
    sacl   temp          ;
    out    temp, IOSR    ;
    and    #0FFFEh       ; reset IO0 – CS low
    sacl   temp          ;
    out    temp, IOSR    ;
    lacl   r_ptr         ; load rolling pointer to accumulator
    add    #sinevals     ; add pointer to table start
    sacl   DACa_ptr      ; to get a pointer for next DAC a sample
    add    #08h          ; add 8 to get to DAC C pointer
    sacl   DACc_ptr
    add    #08h          ; add 8 to get to DAC B pointer
    sacl   DACb_ptr
    add    #08h          ; add 8 to get to DAC D pointer
    sacl   DACd_ptr
    mar    *,ar0         ; set ar0 as current AR

    ; DAC A
    lar    ar0, DACa_ptr ; ar0 points to DAC a sample
    lacl   *             ; get DAC a sample into accumulator
    or     #DACa_control ; OR in DAC A control bits
    sacl   temp          ;
    out    temp, SDTR    ; send data
;------------------------------------------------------------------------------------------
; We must wait for transmission to complete before writing next word to the SDTR.;
TLV5614/04 interface does not allow the use of burst mode with the full packet; rate, as
we need a CLKX –ve edge to clock in last bit before FS goes high again,; to allow SPI
compatibility.
;------------------------------------------------------------------------------------------
    rpt    #016h         ; wait long enough for this configuration
    nop                  ; of MCLK/CLKOUT1 rate

    ; DAC B
    lar    ar0, dacb_ptr ; ar0 points to DAC a sample
    lacl   *             ; get DAC a sample into accumulator
    or     #DACb_control ; OR in DAC B control bits
    sacl   temp          ;
    out    temp, SDTR    ; send data
    rpt    #016h         ; wait long enough for this configuration
    nop                  ; of MCLK/CLKOUT1 rate

; DAC C
    lar    ar0, dacc_ptr ; ar0 points to dac a sample
    lacl   *             ; get DAC a sample into accumulator
    or     #DACc_control ; OR in DAC C control bits
    sacl      temp       ;
    out       temp, SDTR; send data
    rpt       #016h      ; wait long enough for this configuration
    nop                  ; of MCLK/CLKOUT1 rate
```

## APPLICATION INFORMATION

```
; DAC D
   lar       ar0, dacd_ptr; ar0 points to DAC a sample
   lacl   *             ; get DAC a sample into accumulator
   or     #dacd_control ; OR in DAC D control bits
   sacl   temp          ;
   out    temp, SDTR    ; send data

   lacl   r_ptr         ; load rolling pointer to accumulator
   add    #1h           ; increment rolling pointer
   and    #001Fh        ; count 0–31 then wrap back round
   sacl   r_ptr         ; store rolling pointer
   rpt    #016h         ; wait long enough for this configuration
   nop                  ; of MCLK/CLKOUT1 rate

; now take CS high again
   lacl   iosr_stat     ; load acc with iosr status
   or     #0001h        ; set IO0 – CS high
   sacl   temp          ;
   out    temp, IOSR    ;
   clrc   intm          ; re-enable interrupts
   ret                  ; return from interrupt
.end
```

## APPLICATION INFORMATION

### TLV5614 interfaced to MCS®51 microcontroller

#### hardware iInterfacing

Figure 18 shows an example of how to connect the TLV5614 to an MCS®51 Microcontroller. The serial DAC input data and external control signals are sent via I/O Port 3 of the controller. The serial data is sent on the RxD line, with the serial clock output on the TxD line. Port 3 bits 3, 4, and 5 are configured as outputs to provide the DAC latch update (LDAC), chip select (CS) and frame sync (FS) signals for the TLV5614. The active low power down pin (PD) of the TLV5614 is pulled high to ensure that the DACs are enabled.

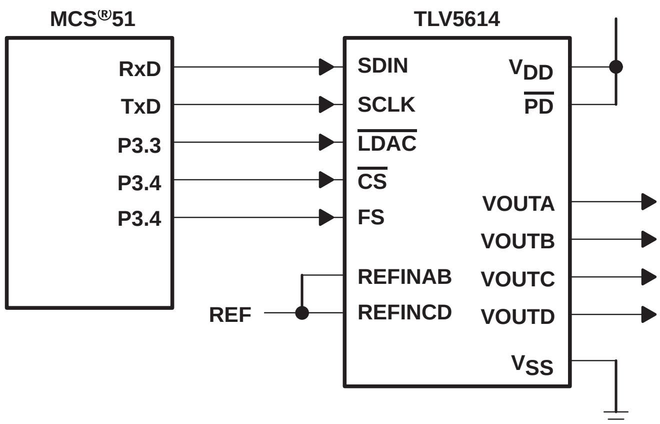



**Figure 18. TLV5614 Interfaced with MCS®51**

#### software

The example is the same as for the TMS320C203 in this datasheet, but adapted for a MCS®51 controller. It generates a differential in-phase (sine) signal between the VOUTA and VOUTB pins, and it's quadrature (cosine) signal as the differential signal between VOUTC and VOUTD.

The on-chip timer is used to generate interrupts at a fixed frequency. The related interrupt service routine pulses LDAC low to update all 4 DACs simultaneously, then fetches and writes the next sample to all 4 DACs. The samples are stored as a look-up table, which describes one full period of a sine wave.

The serial port of the controller is used in Mode 0, which transmits 8 bits of data on RxD, accompanied by a synchronous clock on TxD. Two writes concatenated together are required to write a comlpete word to the TLV5614. The CS and FS signals are provided in the required fashion through control of IO port 3, which has bit addressable outputs.

## APPLICATION INFORMATION

```
;----------------------------------------------------------------------------
; Processor:  80C51
;
; Description:
;
; This program generates a differential in-phase
(sine) on (OUTA–OUTB) ; and it's quadrature (cosine)
 as a differential signal on (OUTC–OUTD).
;
; © 1998, Texas Instruments Inc.
;--------------------------------------------------------------------------------
NAME   GENIQ
MAIN   SEGMENT      CODE
ISR    SEGMENT      CODE
SINTBL SEGMENT      CODE
VAR1   SEGMENT      DATA
STACK  SEGMENT      IDATA
;--------------------------------------------------------------------------------
; Code start at address 0, jump to start
;--------------------------------------------------------------------------------
   CSEG  AT   0
   LJMP   start        ; Execution starts at address 0 on power–up.
;--------------------------------------------------------------------------------
; Code in the timer0 interrupt vector
;--------------------------------------------------------------------------------
   CSEG  AT   0BH
   LJMP   timer0isr    ; Jump vector for timer 0 interrupt is 000Bh
;--------------------------------------------------------------------------------
; Global variables need space allocated
;--------------------------------------------------------------------------------
   RSEG      VAR1
temp_ptr:    DS     1
rolling_ptr: DS     1
;--------------------------------------------------------------------------------;
Interrupt service routine for timer 0 interrupts
;--------------------------------------------------------------------------------
   RSEG      ISR
timer0isr:
   PUSH      PSW
   PUSH      ACC
   CLR       INT1      ; pulse LDAC low
   SETB      INT1      ; to latch all 4 previous values at the same time
                       ; 1st thing done in timer isr => fixed period
   CLR    T0           ; set CS low

   ; The signal to be output on each DAC is a sine function.
   ; One cycle of a sine wave is held in a table @ sinevals
   ; as 32 samples of msb, lsb pairs (64 bytes).
   ; We have ; one pointer which rolls round this table, rolling_ptr,
   ; incrementing by 2 bytes (1 sample) on each interrupt (at the end of
   ; this routine).

   ; The DAC samples are read at an offset to this rolling pointer:
   ; DAC  Function  Offset from rolling_ptr
   ;  A   sine         0
   ;  B   inverse sine 32
   ;  C   cosine       16
   ;  D   inverse cosine48
   MOV    DPTR,#sinevals; set DPTR to the start of the table
                       ; of sine signal values
   MOV    R7,rolling_ptr; R7 holds the pointer
                       ;into the sine table

   MOV    A,R7         ; get DAC A msb
   MOVC   A,@A+DPTR    ; msb of DAC A is in the ACC
```

## APPLICATION INFORMATION

```
    CLR    T1              ; transmit it – set FS low
    MOV    SBUF,A          ; send it out the serial port

    INC    R7              ; increment the pointer in R7
    MOV    A,R7            ; to get the next byte from the table
    MOVC   A,@A+DPTR       ; which is the lsb of this sample, now in ACC
    A_MSB_TX:
    JNB    TI,A_MSB_TX     ; wait for transmit to complete
    CLR    TI              ; clear for new transmit
    MOV    SBUF,A          ; and send out the lsb of DAC A

    ; DAC C next
    ; DAC C codes should be taken from 16 bytes (8 samples) further on
    ; in the sine table – this gives a cosine function
    MOV    A,R7            ; pointer in R7
    ADD    A,#0FH          ; add 15 – already done one INC
    ANL       A,#03FH      ; wrap back round to 0 if > 64
    MOV    R7,A            ; pointer back in R7

    MOVC   A,@A+DPTR       ; get DAC C msb from the table
    ORL    A,#01H          ; set control bits to DAC C address
A_LSB_TX:
    JNB    TI,A_LSB_TX     ; wait for DAC A lsb transmit to complete
    SETB   T1              ; toggle FS
    CLR T1
    CLR    TI              ; clear for new transmit
    MOV    SBUF,A          ; and send out the msb of DAC C
    INC    R7              ; increment the pointer in R7
    MOV    A,R7            ; to get the next byte from the table
    MOVC   A,@A+DPTR       ; which is the lsb of this sample, now in ACC
C_MSB_TX:
    JNB    TI,C_MSB_TX     ; wait for transmit to complete
    CLR    TI              ; clear for new transmit
    MOV    SBUF,A          ; and send out the lsb of DAC C

    ; DAC B next
    ; DAC B codes should be taken from 16 bytes (8 samples) further on
    ; in the sine table – this gives an inverted sine function
    MOV    A,R7            ; pointer in R7
    ADD       A,#0FH       ; add 15 – already done one INC
    ANL    A,#03FH         ; wrap back round to 0 if > 64
    MOV    R7,A            ; pointer back in R7

    MOVC   A,@A+DPTR       ; get DAC B msb from the table
    ORL       A,#02H       ; set control bits to DAC B address

C_LSB_TX:
    JNB    TI,C_LSB_TX     ; wait for DAC C lsb transmit to complete
    SETB   T1              ; toggle FS
    CLR    T1
    CLR    TI              ; clear for new transmit
    MOV    SBUF,A          ; and send out the msb of DAC B

    ; get DAC B LSB
    INC    R7              ; increment the pointer in R7
    MOV    A,R7            ; to get the next byte from the table
    MOVC   A,@A+DPTR       ; which is the lsb of this sample, now in ACC

B_MSB_TX:
    JNB    TI,B_MSB_TX     ; wait for transmit to complete
    CLR    TI              ; clear for new transmit
    MOV       SBUF,A       ; and send out the lsb of DAC B

    ; DAC D next
    ; DAC D codes should be taken from 16 bytes (8 samples) further on
    ; in the sine table – this gives an inverted cosine function
```

## APPLICATION INFORMATION

```
    MOV    A,R7           ; pointer in R7
    ADD    A,#0FH         ; add 15 – already done one INC
    ANL       A,#03FH     ; wrap back round to 0 if > 64
    MOV    R7,A           ; pointer back in R7
    MOVC   A,@A+DPTR      ; get DAC D msb from the table
    ORL    A,#03H         ; set control bits to DAC D address

B_LSB_TX:
    JNB    TI,B_LSB_TX    ; wait for DAC B lsb transmit to complete
    SETB   T1             ; toggle FS
    CLR    T1
    CLR    TI ; clear for new transmit
MOV    SBUF,A             ; and send out the msb of DAC D

    INC    R7             ; increment the pointer in R7
    MOV    A,R7           ; to get the next byte from the table
    MOVC   A,@A+DPTR      ; which is the lsb of this sample, now in ACC
D_MSB_TX:
    JNB    TI,D_MSB_TX    ; wait for transmit to complete
    CLR    TI             ; clear for new transmit
    MOV    SBUF,A         ; and send out the lsb of DAC D

    ; increment the rolling pointer to point to the next sample
    ; ready for the next interrupt
    MOV    A,rolling_ptr
    ADD    A,#02H         ; add 2 to the rolling pointer
    ANL    A,#03FH        ; wrap back round to 0 if > 64
    MOV    rolling_ptr,A  ; store in memory again
D_LSB_TX:
    JNB    TI,D_LSB_TX    ; wait for DAC D lsb transmit to complete
    CLR    TI             ; clear for next transmit
    SETB   T1             ; FS high
    SETB   T0             ; CS high
    POP    ACC
    POP    PSW
    RETI

;--------------------------------------------------------------------------------
; Stack needs definition
;--------------------------------------------------------------------------------
    RSEG  STACK
    DS    10h             ; 16 Byte Stack!
;--------------------------------------------------------------------------------
; Main program code
;--------------------------------------------------------------------------------
    RSEG   MAIN
start:
    MOV    SP,#STACK–1    ; first set Stack Pointer
    CLR A
    MOV    SCON,A         ; set serial port 0 to mode 0
    MOV    TMOD,#02H      ; set timer 0 to mode 2 – auto–reload
    MOV    TH0,#038H      ; set TH0 for 5kHs interrupts
    SETB   INT1           ; set LDAC = 1
    SETB   T1             ; set FS = 1
    SETB   T0             ; set CS = 1
    SETB   ET0            ; enable timer 0 interrupts
    SETB   EA             ; enable all interrupts
    MOV    rolling_ptr,A  ; set rolling pointer to 0
    SETB   TR0            ; start timer 0
always:
    SJMP   always         ; while(1) !
    RET
;--------------------------------------------------------------------------------
; Table of 32 sine wave samples used as DAC data
;--------------------------------------------------------------------------------
    RSEG   SINTBL
```

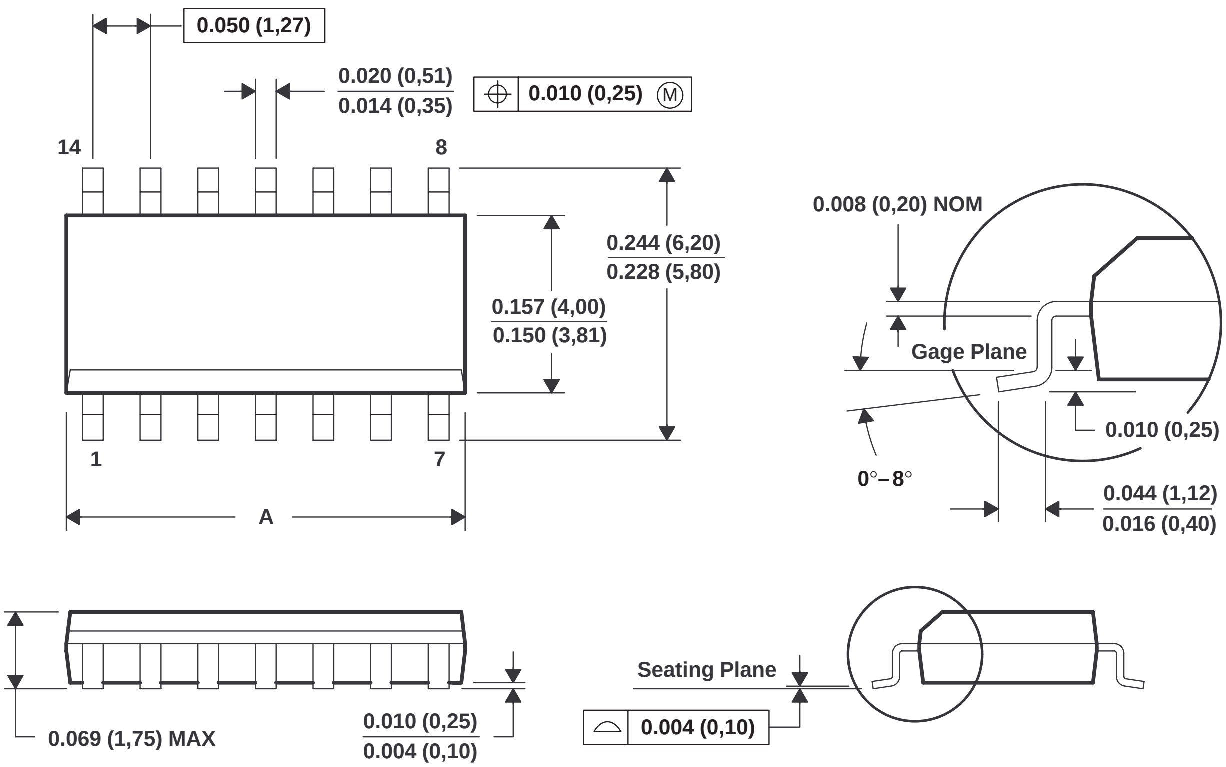
## APPLICATION INFORMATION

```
sinevals:
    DW     01000H
    DW     0903EH
    DW     05097H
    DW     0305CH
    DW     0B086H
    DW     070CAH
    DW     0F0E0H
    DW     0F06EH
    DW     0F039H
    DW     0F06EH
    DW     0F0E0H
    DW     070CAH
    DW     0B086H
    DW     0305CH
    DW     05097H
    DW     0903EH
    DW     01000H
    DW     06021H
    DW     0A0E8H
    DW     0C063H
    DW     040F9H
    DW     080B5H
    DW     0009FH
    DW     00051H
    DW     00026H
    DW     00051H
    DW     0009FH
    DW     080B5H
    DW     040F9H
    DW     0C063H
    DW     0A0E8H
    DW 06021H
END
```

## MECHANICAL DATA

**D (R-PDSO-G**)** **PLASTIC SMALL-OUTLINE PACKAGE**

**14 PIN SHOWN**




| DIM \ PINS ** | 8 | 14 | 16 |
|---|---|---|---|
| A MAX | 0.197 (5,00) | 0.344 (8,75) | 0.394 (10,00) |
| A MIN | 0.189 (4,80) | 0.337 (8,55) | 0.386 (9,80) |

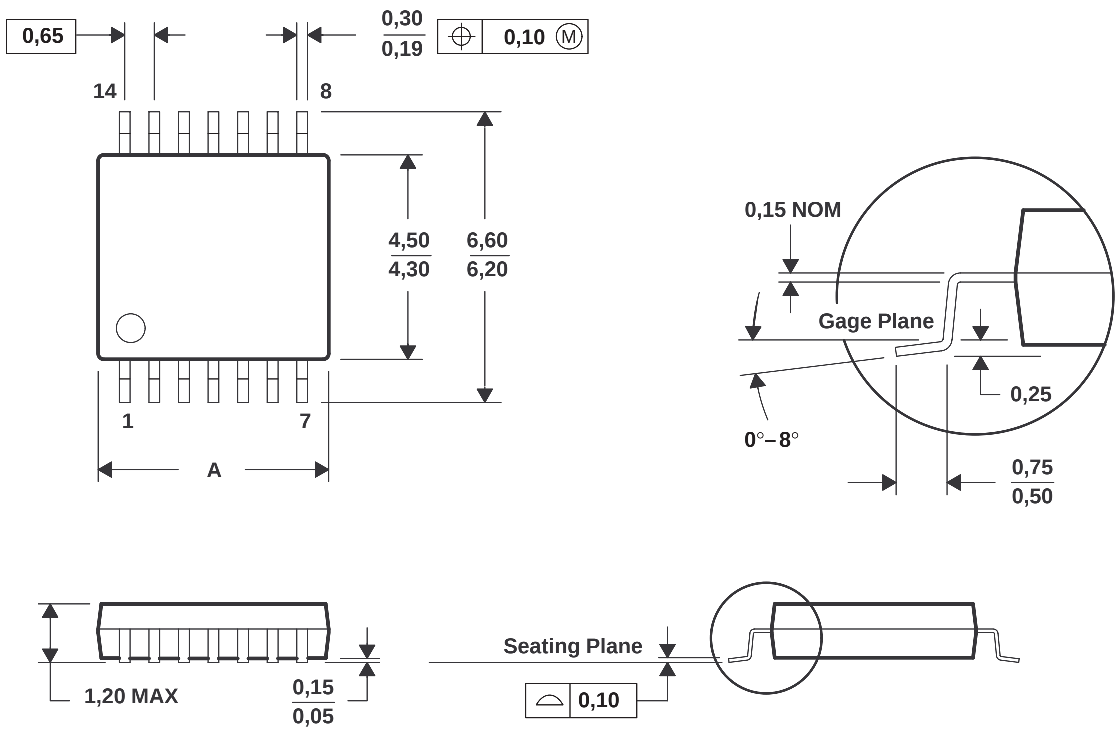
4040047 / D 10/96

NOTES: A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion, not to exceed 0.006 (0,15).
D. Falls within JEDEC MS-012

## MECHANICAL DATA

**PW (R-PDSO-G**)** **PLASTIC SMALL-OUTLINE PACKAGE**

**14 PIN SHOWN**




| PINS ** / DIM | 8 | 14 | 16 | 20 | 24 | 28 |
|---|---|---|---|---|---|---|
| A MAX | 3,10 | 5,10 | 5,10 | 6,60 | 7,90 | 9,80 |
| A MIN | 2,90 | 4,90 | 4,90 | 6,40 | 7,70 | 9,60 |

**4040064 / E 08/96**

NOTES:
- A. All linear dimensions are in millimeters.
- B. This drawing is subject to change without notice.
- C. Body dimensions do not include mold flash or protrusion not to exceed 0,15.
- D. Falls within JEDEC MO-153

## IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.