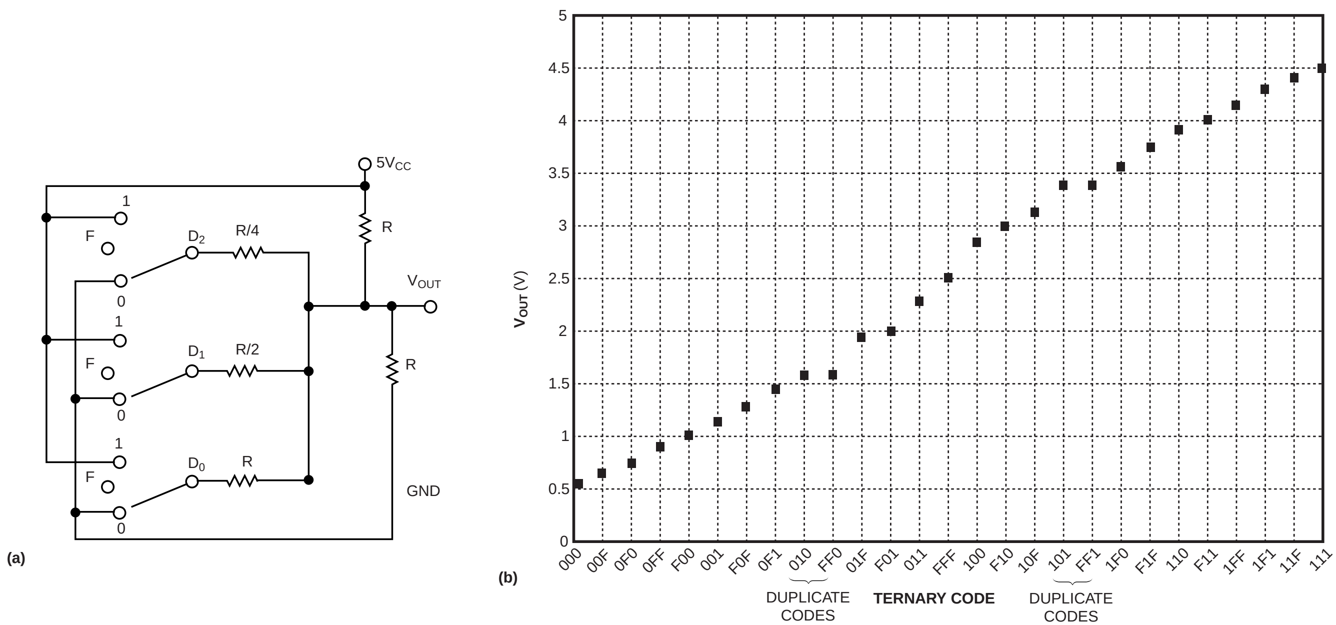
Technique increases low-cost DAC's resolution

Jeremy Dean, Thomson-Thorn Missile Electronics Ltd, Basingstoke, UK

Cost-sensitive μC applications often employ resistor chains to implement crude DACs. You can extend this method by exploiting the way in which many μCs allow individual output pins to be set to either low ("0''), high ("1''), or floating ("F") states. A converter can thus respond to ternary rather than binary codes.

The resistive network in **Figure 1a** has three inputs. Allowing each input to be either 0, 1, or F results in the transfer characteristic in **Figure 1b**. Allowing for two duplicate cases results in 25 distinct output levels. Thus, the technique achieves roughly 4½ bits of resolution while using only three pins of a μC and five resistive elements. The transfer characteristic is symmetrical about midrail and does not extend to the supply rails, making it inherently suitable for use in

FIGURE 1



Providing three possible inputs—1, 0, and F (floating)—to the resistor network (a) produces a ternary-code transfer characteristic (b).

single-supply applications. Note that the characteristic is nonlinear, which shouldn't matter for many applications.

A practical circuit implementation uses a single-inline array to form the resistive network (**Figure 2**). The PIC16C84 (Microchip Technology, www.microchip.com) code in **Listing 1** uses a look-up table to convert binary inputs to the required ternary outputs. (You can download this **listing** and the related look-up table from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2189.)

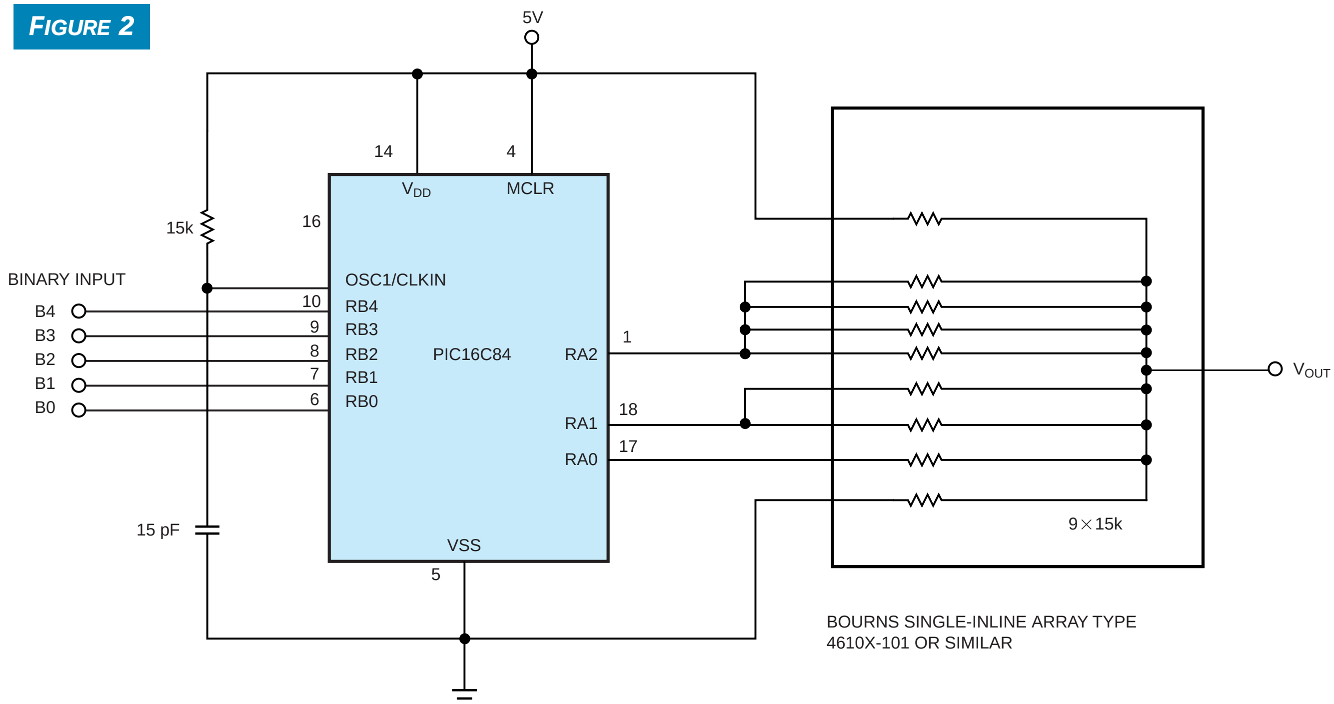
You can expand or contract this ternary technique within reason. For example, using four µC pins gives 75 distinct output levels, and even just two µC pins gives seven levels. By saving pins, the technique is ideally suited for use with the PIC12C50X family, which has very limited I/O in an eight-pin package. (DI #2189)

To Vote For This Design, Circle No. 528

LISTING 1—PIC16C84 DAC CODE

```
; Title:    c:\edn\cheapdac\code_v1.asm
; Author:   Jes Dean
; Dated:    20/10/97
; Function: Program continually loops, reading 5 bit binary i/p
;           from RB4-0, and outputting ternary codes on RA2-0
;           suitable for driving resistor array as demonstrator.

        LIST    P=16C84

PC              equ     0x02 ; Relevant system registers...
STATUS          equ     0x03
PORTA           equ     0x05
PORTB           equ     0x06
TRISA           equ     0x85

temp            equ     0x10 ; Temporary storage.

        org     0x00
        goto    main

        org     0x10
; For brevity, no explicit initialisation done. PORTA, PORTB initialise
; as inputs with no weak pull-ups, and all interrupts are disabled.
main    movf    PORTB,0         ; Read PORTB
        movwf   temp            ; and store.

        sublw   26
        btfsc   STATUS,0        ; IF 0=<w=<26
        goto    inrange         ; THEN temp is in correct range.
        movlw   26              ; ELSE
        movwf   temp            ; Limit temp to 26.

inrange movf    temp,0          ; Copy temp to w
        call    ra_lut          ; and call look up table.
        bsf     STATUS,5        ; Enable addressing of TRISA (in Bank 1)
        movwf   TRISA           ; Set RA2-0 as inputs or outputs accordingly.
                                ;   (Note RA4,3 will be set spuriously.)
        bcf     STATUS,5        ; Disable addressing of TRISA (in Bank 1)
        movwf   temp
        swapf   temp,0          ;
        movwf   PORTA           ; Put relevant data on pins set as outputs.
                                ;   (Note RA4,3 may be set spuriously.)

        movlw   0xff            ; Delay loop to allow value to
        movwf   temp            ; settle for measurement purposes.
loop1   decfsz  temp,1
        goto    loop1

        goto    main            ; Repeat eternally.
```

FIGURE 2



A simple DAC demonstration circuit comprises a PIC16C84 µC and a single-inline-resistor array.