

Interfacing the AD7549 to the MCS-48 and MCS-51 Microcomputer Families

by Mike Curtin

The AD7549 is the industry's first monolithic dual 12-bit D/A converter. Not only that, but because of the data loading structure, it is housed in a space-saving 20-pin 0.3" wide package. The data for each DAC is contained in three 4-bit nibbles. Table I shows how the digital control inputs load the internal registers. The dedicated CLR line is especially useful in system reset and autocalibration routines while $\overline{\text{UPD}}$ allows both DACs to be updated together from their respective input registers. The AD7549 data sheet gives further information on the device and it is recommended that the user consult this where necessary.

The applications of the AD7549 are many and varied. They include volume and balance control in audio systems, cutoff frequency and Q factor control in programmable filters and frequency and amplitude control in programmable oscillators. As well as these audio based applications,

the AD7549 can be used in general to replace two discrete DACs with a dual DAC in a small package. One particular example of this is when the DACs are used as analog output ports in microcomputer based control systems. This application note examines such systems and deals with the hardware and software interfacing needs.

CLR	$\overline{\text{UPD}}$	$\overline{\text{CS}}$	WR	A2	A1	A0	FUNCTION
0	X	X	1	X	X	X	No data transfer.
0	1	1	X	X	X	X	No data transfer.
1	X	X	X	X	X	X	All registers cleared.
0	1	0	$\overline{\text{L}}$	0	0	0	DAC A LOW NIBBLE REGISTER loaded from Data Bus.
0	1	0	$\overline{\text{L}}$	0	0	1	DAC A MID NIBBLE REGISTER loaded from Data Bus.
0	1	0	$\overline{\text{L}}$	0	1	0	DAC A HIGH NIBBLE REGISTER loaded from Data Bus.
0	1	0	$\overline{\text{L}}$	0	1	1	DAC A Register loaded from Input Registers.
0	1	0	$\overline{\text{L}}$	1	0	0	DAC B LOW NIBBLE REGISTER loaded from Data Bus.
0	1	0	$\overline{\text{L}}$	1	0	1	DAC B MID NIBBLE REGISTER loaded from Data Bus.
0	1	0	$\overline{\text{L}}$	1	1	0	DAC B HIGH NIBBLE REGISTER loaded from Data Bus.
0	1	0	$\overline{\text{L}}$	1	1	1	DAC B Register loaded from Input Registers.
0	0	1	$\overline{\text{L}}$	X	X	X	DAC A, DAC B Registers updated simultaneously from Input Registers.

NOTE: X = Don't Care

Table I. AD7549 Truth Table

Microcomputers differ from microprocessors in that they contain not only an on-chip CPU but also various combinations of RAM, ROM, I/O ports, timers, etc., which enables the user to configure a complete control system with a minimum amount of hardware. It is, therefore, hardly surprising that the number of microcomputers used in instrumentation and control is ever increasing. Two of the most popular microcomputer families are the MCS-48 and the newer MCS-51, both of which are manufactured by Intel. Interfacing the AD7549 to these systems requires little external hardware and keeps system component count to a minimum.

MCS-48 INTERFACE

The various members of the MCS-48 family differ only in their internal memory capabilities. In the following discussion the 8048 microcomputer is representative of the

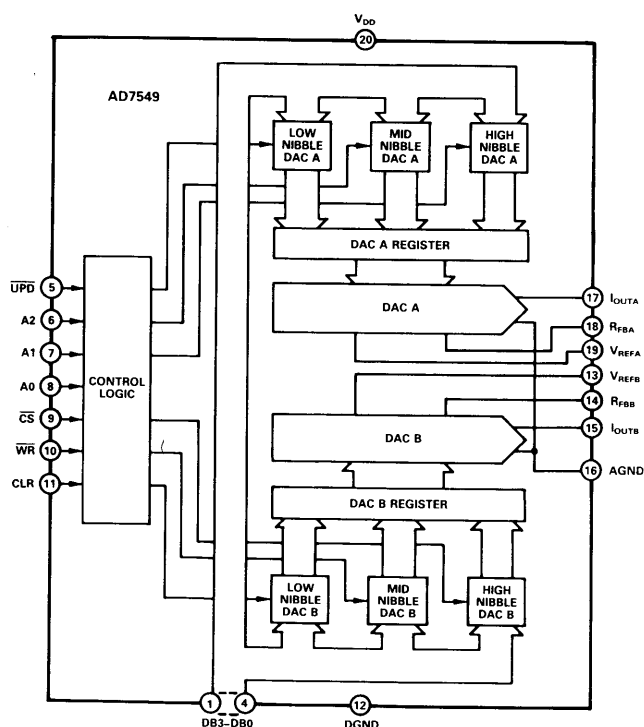


Figure 1. AD7549 Functional Diagram

complete family. The 8048 contains a set of specific instructions for communicating with the 8243 I/O expander. These are the MOVD, ANLD, ORLD instructions.

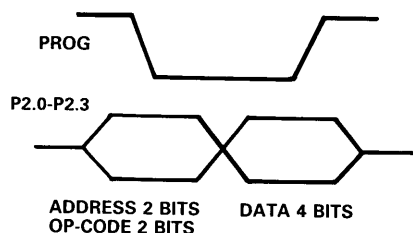


Figure 2. MCS-48 Input/Output Expander Timing

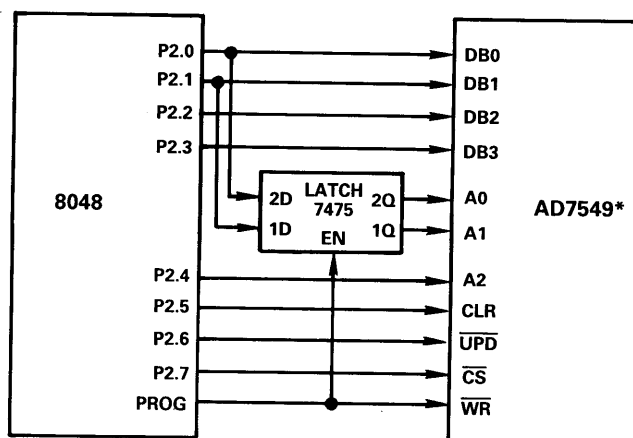
MOVD reads or writes data directly between the micro-computer's accumulator and one of the four output ports on the 8243 I/O expander. ORLD and ANLD perform the logical OR and logical AND functions between the accumulator and an output port, placing the result in the output port. Figure 2 shows the timing for the Output Expander instructions. All communication with the 8048 occurs over the lower half of Port 2 (P2.0–P2.3). Timing control is provided by PROG. Each transfer consists of two 4-bit nibbles. The first contains the op code and port address and the second contains the actual 4-bits of data. A high to low transition of the PROG line indicates that address is present, while a low to high indicates the presence of data. Because of the similarity in structure between the AD7549 and the 8243 it is possible to use the 8243 instructions to program the AD7549.

P2.0	P2.1	8243 Port	AD7549 Register
0	0	P4	LOW NIBBLE REGISTER
0	1	P5	MID NIBBLE REGISTER
1	0	P6	HIGH NIBBLE REGISTER
1	1	P7	DAC REGISTER

Table II. AD7549 Register Selection

Table II shows how the registers are selected. As an example, the instruction "MOVD P4, A" sets P2.0 and P2.1 to zeros in the first nibble of the instruction (Figure 2). The negative going edge of PROG latches 0 0 to A0 and A1 causing the LOW NIBBLE REGISTER to be loaded with the bottom 4 bits of A. Note that the AD7549 A2 pin is not shown in Table II. This effectively selects DAC A or DAC B and would be set appropriately prior to the MOVD instruction. Figure 3 shows the interface circuit for the MCS-48 to the AD7549.

Data inputs DB3–DB0 connect directly to P2.3–P2.0. Address lines A0, A1 are obtained by latching P2.0, P2.1 on the negative-going edge of PROG. All other AD7549 control lines are port outputs on the 8048. Table III is a listing of OUTLD, a routine for loading the dual DAC. To load DAC A with UVW (HEX) the routine initially selects the de-



*LINEAR CIRCUITRY OMITTED FOR CLARITY

Figure 3. Interface Circuit for the MCS-48 to AD7549

vice, and clears all registers. Then, using the MOVD instruction, each nibble register is loaded with its correct 4 bits of data. Finally the DAC A Register is loaded with the contents of the 3 nibble registers. The routine then goes through a similar procedure to load DAC B with XYZ (HEX). When the high speed (11MHz) version of the 8048 is used, it takes 57μs to load both DACs.

0000	OUTLD:	MOV	A, #60H	Bring $\overline{CS}$ low to select
02		OUTL	P2, A	device and set CLR
				high to clear all DAC
				registers
03		MOV	A, #40H	Disable CLR, Set
05		OUTL	P2, A	A2 = 0 to select DAC A
06		MOV	A, #0VWH	Load DAC A LOW
				NIBBLE REGISTER
08		MOVD	P4, A	with W (HEX)
09		SWAP	A	Load DAC A MID
				NIBBLE REGISTER
0A		MOVD	P5, A	with V (HEX)
0B		MOV	A, #0UH	Load DAC A HIGH
				NIBBLE REGISTER
0D		MOVD	P6, A	with U (HEX)
0E		MOVD	P7, A	Load DAC A
				REGISTER with UVW
				(HEX)
0F		MOV	A, #50H	Set A2 = 1
				to select DAC B
11		OUTL	P2, A	
12		MOV	A, #0YZH	Load DAC B LOW
				NIBBLE REGISTER
14		MOVD	P4, A	with Z (HEX)
15		SWAP	A	Load DAC B MID
				NIBBLE REGISTER
16		MOVD	P5, A	with Y (HEX)
17		MOV	A, #0XH	Load DAC B HIGH
				NIBBLE REGISTER
19		MOVD	P6, A	with X
1A		MOVD	P7, A	Load DAC B
				REGISTER with XYZ
				(HEX)
1B		MOV	A, #C0H	Set $\overline{CS}$ high to
				deselect DAC
1D		OUTL	P2, A	

Table III. 8048 Program Listing

MCS-51 INTERFACE

In the following text, the term "8051" is used to generically refer to all members of the MCS-51 family. Like the MCS-48 family, the various members differ mainly in their internal memory capabilities (the 8052/8032 also has an extra 16-bit timer/counter). In comparison to the 8048, the 8051 offers more speed, more input-output pins, and more memory in addition to a full duplex serial I/O port. There is no facility on the 8051 to expand I/O ports in the same manner as the 8048. However, it does have features which make an AD7549 interface very simple. In particular, the 8051's Boolean processing instructions provide direct bit handling (i.e., single bit addressing and updating). Individual bits can be set, cleared or complemented with the 2-byte instructions SETB, CLR or CPL. In addition, bits can be moved to and from the carry flag with the MOV instruction, and logical ANL and ORL functions can be performed between the carry and the addressed bit. With this bit handling facility, it is possible to interface to the AD7549 without any external hardware.

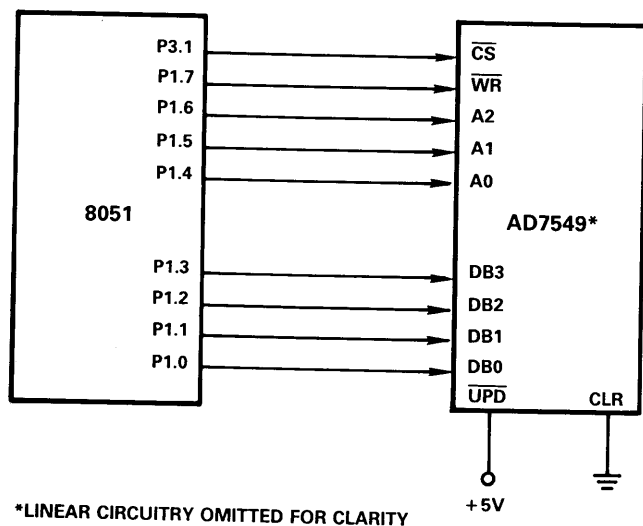


Figure 4. MCS-51 to AD7549 Interface

Figure 4 shows the interface circuit and Table IV lists the software routine DACLOAD which loads both DACs. The data to be loaded into the DACs is stored in data memory locations 20 to 23. Table V shows how this is organized. DACLOAD programs DAC A with UVW (HEX) and DAC B with XYZ (HEX). P3.1 is the AD7549 CS. This is activated at the beginning of the routine and brought high at the end. Port lines P1.4–P1.6 control the register addressing. The routine initially sets up the appropriate nibble register address in 8051 register R2. Then, the ADDR5 subroutine formats this and loads it to the device address lines. The DATA subroutine places the correct data on the device lines DB3–DB0 and loads the appropriate register by strobing WR (P1.7) low. The next AD7549 nibble register address is then set up by incrementing R2. When the three nibble registers are loaded the 12-bits of data are transferred to the DAC register by setting up the DAC register address and strobing WR low.

0100	DACLOAD:	CLR	P3.1	Bring CS low to select device
102		MOV	R2, #00	Load 1st register address into R2
104		LCAL	ADDRS	Register address loaded to DAC
107		MOV	R0, #21	
109		MOV	A, @R0	
10A		LCAL	DATA	Data (W) loaded into register
10D		INC	R2	Next register address setup
10E		LCAL	ADDRS	Register address loaded to DAC
111		MOV	A, @R0	
112		SWAP	A	
113		LCAL	DATA	Data (V) loaded into register
116		INC	R2	Next register address setup
117		LCAL	ADDRS	Register address loaded to DAC
11A		DEC	R0	
11B		MOV	A, @R0	
11C		LCAL	DATA	Data (U) loaded into register
11F		INC	R2	DAC A register address setup
120		LCAL	ADDRS	DAC A register address loaded to DAC
123		CLR	P1.7	Bring WR low
125		SETB	P1.7	Bring WR high. Data (UVW) has now been loaded to DAC A
127		INC	R2	Next register address setup
128		LCAL	ADDRS	Register address loaded to DAC
12B		INC	R0	
12C		INC	R0	
12D		INC	R0	
12E		MOV	A @R0	
12F		LCAL	DATA	Data (Z) loaded into register
132		INC	R2	Next register address setup
133		LCAL	ADDRS	Register address loaded to DAC
136		MOV	A, @R0	
137		SWAP	A	
138		LCAL	DATA	Data (Y) loaded into register
13B		INC	R2	Next register address setup
13C		LCAL	ADDRS	Register address loaded to DAC
13F		DEC	R0	
140		MOV	A @R0	
141		LCAL	DATA	Data (X) loaded into register
144		INC	R2	DAC B register address setup
145		LCAL	ADDRS	DAC B register address loaded to DAC
148		CLR	P1.7	Bring WR low
14A		SETB	P1.7	Bring WR high. Data (XYZ) has now been loaded to DAC B
14C		SETB	P3.1	Bring CS high to deselect device
0200	ADDRS:	MOV	A, R2	This subroutine takes the appropriate register address held in R2, formats it and loads it out to the device.
201		SWAP	A	
202		ORL	A, #80	
204		MOV	P1, A	
206		RET		
0210	DATA:	ANL	A, #0F	This subroutine transfers the data nibble in A to the device data bus and strobes the WR line low to load the appropriate register.
212		ORL	P1, A	
214		CLR	P1.7	
216		SETB	P1.7	
218		RET		

Table IV. 8051 Program Listing

When the 12MHz version of the 8051 is used, the program loads the AD7549 in 140μs. The amount of program memory used is 93 bytes. Note that the device architecture allows interfacing using only nine of the 8051's thirty-two input/output lines, so that in large control systems, the DACs do not unduly overburden the 8051 I/O capability. The system is easily expanded by taking extra I/O lines and using them as CS inputs for successive devices.

Memory Location	Contents
20	0U
21	VW
22	0X
23	YZ

Table V. Data to be Loaded to the AD7549