ADE-207-038

HD81803 ADPCM Transcoder Conforming to G.721

HITACHI

Preliminary
Jan. 1990

Description

The HD81803 is a single-chip ADPCM (Adaptive Differential Pulse Code Modulation) Transcoder conforming to the CCITT Recommendation G.721. It can compress and expand voice data by ADPCM transcoding, and is suitable for high-quality voice recording and playback systems. Applications include ISDN services, voice mail, and private broadcasting.

Converting data from PCM code to ADPCM code reduces the memory needed to store voice data by half the size. Also, since the HD81803 conforms to G.721, it can be used in ISDN line terminal equipment.

The HD81803 can be directly connected to a PCM CODEC, and can convert a 64-kbits/µ-law or A-law PCM channel to and from a 32-kbits/s ADPCM channel.

Features

- Full conformity with CCITT Recommendation G.721
- ADPCM transcoding ensures speech data accuracy
- 64-kbits/s channel can be compressed to 32 kbits/s (Half-Duplex)
- Serial or parallel input/output
- Parallel input/output allows digital sequence tests specified by the CCITT
- Parallel input/output width can be selected as 4, 8, or 16 bits (suitable for a 125-, 250-, or 500-µs microprocessor I/O cycle)
- Through functions are available to input/output specific waveforms without ADPCM transcoding
- A-law or µ-law PCM can be input/output
- Direct connection to a PCM CODEC is possible
- Built-in interface for 8- and 16-bit microprocessors
- Operating mode can be controlled by microprocessor
- +5 V power supply

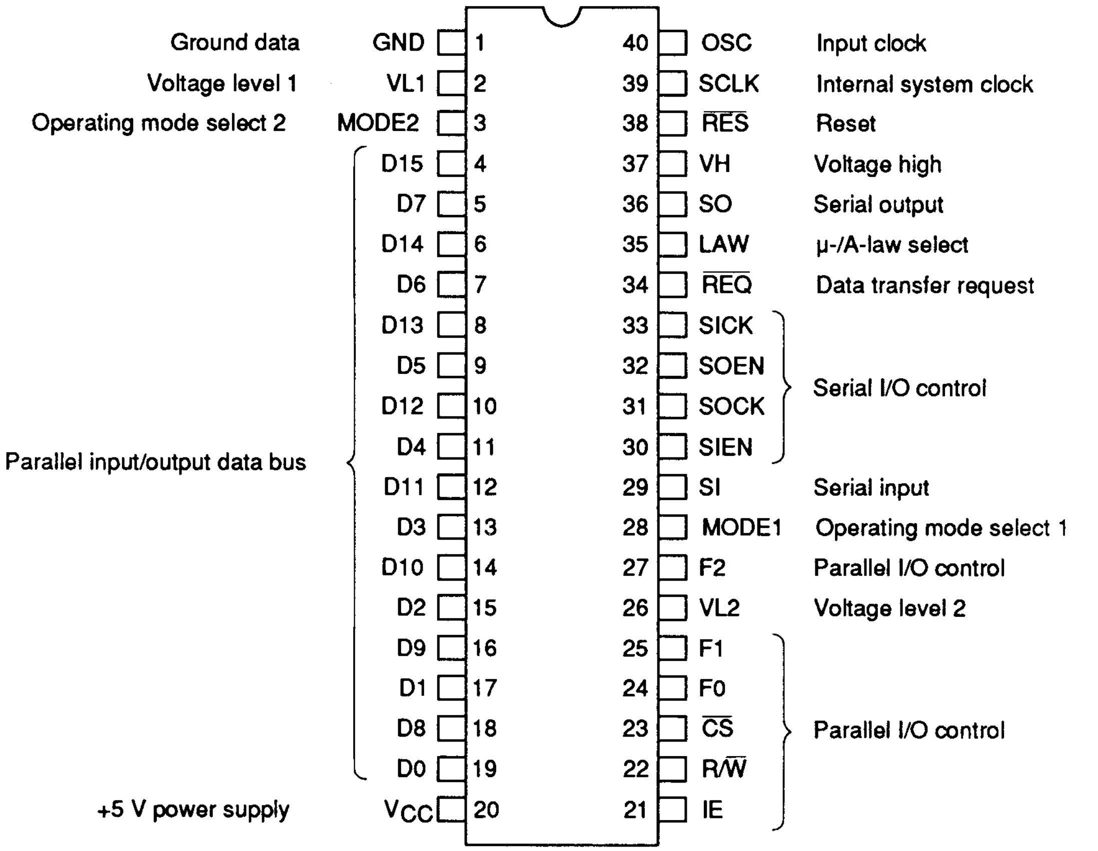
	PCM data	ADPCM data
Encoder	Serial input	Serial or parallel output
Decoder	Serial or parallel output	Serial or parallel input

(Through and test functions are not included.)

Specifications

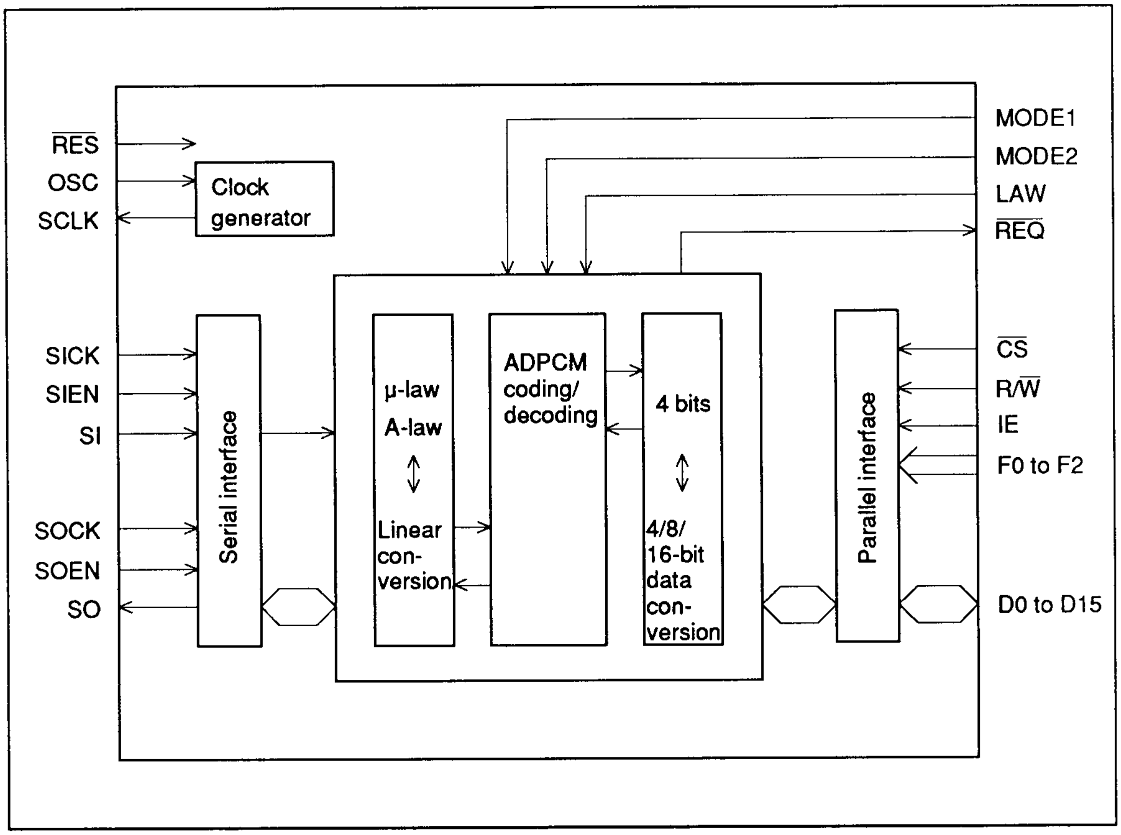
Item	Details
Coding process	CCITT recommendation G.721 ADPCM
Sampling frequency	8 kHz
Code length	4 bits
Data rate	32 kbits/sec
Serial interface (PCM CODEC interface)	(1) µ-law PCM CODEC (ex. HD44238C/HD44278P)
	(2) A-law PCM CODEC (ex. HD44237C/HD44277P)
Parallel interface (microprocessor interface)	(1) 8-bit microprocessor
	(2) 16-bit microprocessor
Supplementary functions	(1) Serial communication (2) High-quality waveform input/output (through function) (3) Digital sequence test (CCITT Recommendation G.721)
Fabrication Process	CMOS 1.3 µm
Power dissipation	150 mW (typ)
Power supply voltage	+5 V
Package	40-pin plastic DIP, 44-pin PLCC

Pin Assignments



(Top view)

Block Diagram



Electrical Characteristics

Absolute Maximum Ratings

Item	Symbol	Value	Unit	Note
Power supply voltage	V_{CC}	–0.3 to +7.0	V	
Input voltage	V_{in}	–0.3 to V_{CC} +0.3	V	
Operating temperature	T_{opr}	–20 to +75	°C	
Storage temperature	T_{stg}	–55 to +125	°C	

DC Characteristics (V_{CC} = 5.0 V ±5%, GND = 0 V, T_a = –20 to +75°C; unless otherwise specified)

Item	Pins	Symbol	Min	Typ	Max	Unit	Test Conditions
Input high voltage	OSC	V_{IH}	2.4	—	V_{CC} + 0.3	V	
	IE, SICK, SOCK		2.4	—	V_{CC} + 0.3		
	$\overline{\text{RES}}$		V_{CC} × 0.7	—	V_{CC} + 0.3		
	Other input pins		2.2	—	V_{CC} + 0.3		
Input low voltage	OSC	V_{IL}	–0.3	—	0.4	V	
	IE, SICK, SOCK		–0.3	—	0.4		
	$\overline{\text{RES}}$		–0.3	—	0.4		
	Other input pins		–0.3	—	0.4		
Input leakage current	IE, R/$\overline{\text{W}}$, $\overline{\text{CS}}$, F0 to F2, SI, SIEN, SOCK, SOEN, SICK, $\overline{\text{RES}}$, OSC	$\lvert I_{IL} \rvert$	—	—	10	μA	V_{in} = 0.4 to 2.4 V
High-impedance leakage current (off state)	D0 to D15, SO	$\lvert I_{TS} \rvert$	—	—	10	μA	V_{in} = 0.4 to 2.4 V
Open drain current (off state)	$\overline{\text{REQ}}$, MODE1, MODE2, LAW	$\lvert I_{LOH} \rvert$	—	—	10	μA	V_{in} = 0.4 to 2.4 V
Output high voltage	D0 to D15, SO	V_{OH}	2.4	—	—	V	$-I_{OH}$ = 400 μA
Output low voltage	D0 to D15, SO, $\overline{\text{REQ}}$	V_{OL}	—	—	0.8	V	I_{OL} = 1.6 mA
Input capacitance	All input pins	Cin	—	—	12.5	pF	V_{in} = 0V, f = 1 MHz, T_a = 25°C
Current dissipation		I_{CC}	—	30	100	mA	No load on output

AC Characteristics (V_{CC} = 5.0 V ±5%, GND = 0 V, T_a = –20 to +75°C unless otherwise specified)

System Clock

Item	Symbol	Min	Typ	Max	Unit	Test Conditions
Clock (OSC) period	ϕ_{cyc}	30.0	31.25	32.5	ns	See Figure 1
Clock (OSC) pulse width	ϕ_{WH}	10	—	—	ns	
	ϕ_{WL}	10	—	—	ns	
Clock (OSC) rise time	ϕ_r	—	—	5	ns	
Clock (OSC) fall time	ϕ_f	—	—	5	ns	

Reset Timing

Item	Symbol	Min	Typ	Max	Unit	Test Conditions
Power-on reset time	t_{RC}	20	—	—	ms	See Figure 2
Reset pulse width	t_{RST}	1.0	—	—	µs	

Serial I/O Timing

Item	Symbol	Min	Typ	Max	Unit	Test Conditions
Clock (SICK, SOCK) period	S_{cyc}	0.4	—	10.0	µs	See Figure 3
Clock (SICK, SOCK) pulse width	S_{WH}	100	—	—	ns	
	S_{WL}	100	—	—	ns	
Clock (SICK, SOCK) rise time	S_r	—	—	20	ns	
Clock (SICK, SOCK) fall time	S_f	—	—	20	ns	
Serial input data setup time	t_{SDS}	50	—	—	ns	
Serial input data hold time	t_{SDH}	50	—	—	ns	
Serial output data delay time	t_{SDD}	—	—	80	ns	
Enable delay time	t_{ED}	50	—	—	ns	
Enable setup time	t_{ES}	50	—	—	ns	

Parallel I/O Timing

Item	Symbol	Min	Typ	Max	Unit	Test Conditions
IE period	t_{cyc}	0.5	—	—	µs	See figure 4
IE pulse width	t_{WH}	210	—	5000	ns	
	t_{WL}	210	—	—	ns	
IE rise time	t_r	—	—	20	ns	
IE fall time	t_f	—	—	20	ns	
CS setup time	t_{CS}	60	—	—	ns	
CS hold time	t_{CH}	10	—	—	ns	
Input data setup time	t_{DSW}	60	—	—	ns	
Input data hold time	t_{DHW}	10	—	—	ns	
Output data delay time	t_{DDR}	—	—	150	ns	
Output data hold time	t_{DHR}	20	—	—	ns	

Timing Diagrams

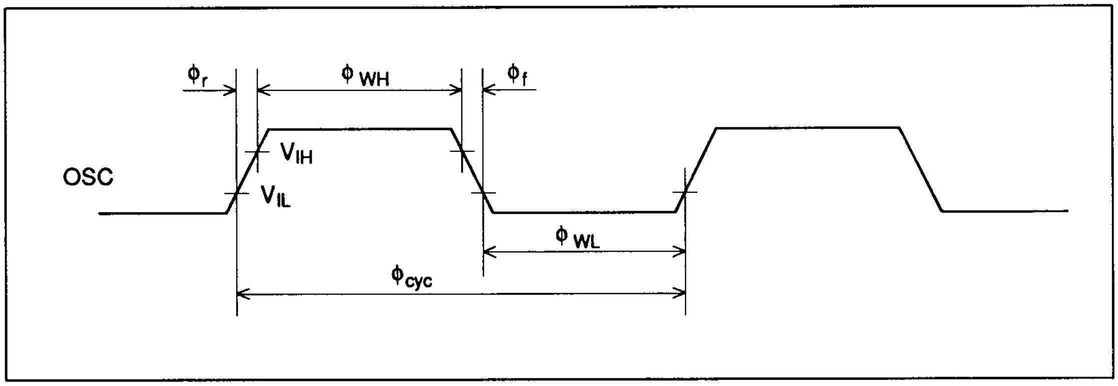
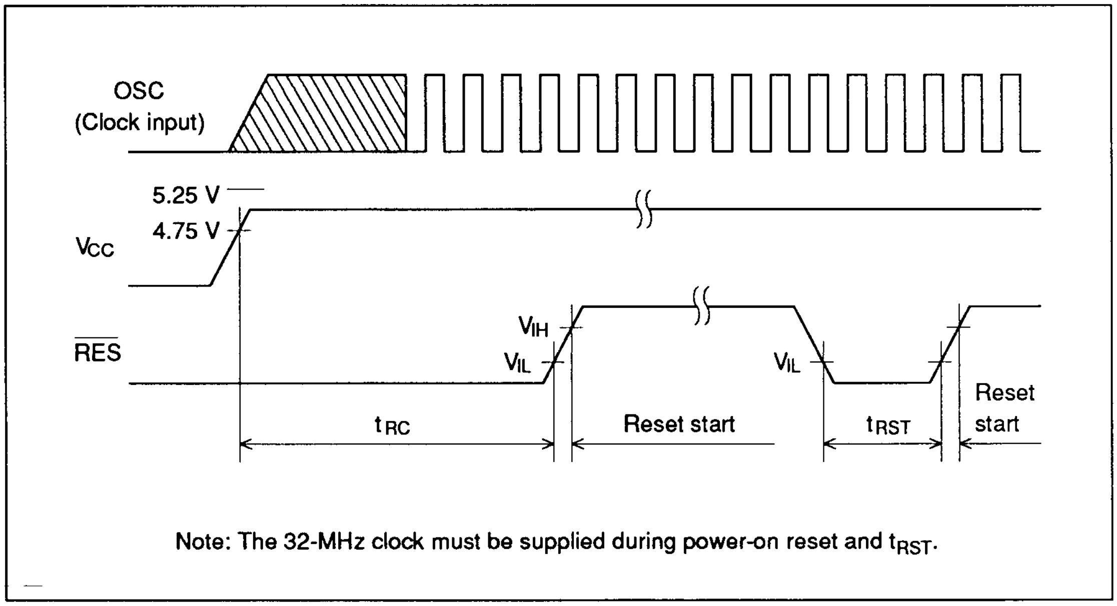


Figure 1 Clock Input Waveform



Note: The 32-MHz clock must be supplied during power-on reset and t_{RST}.

Figure 2 Reset Timing

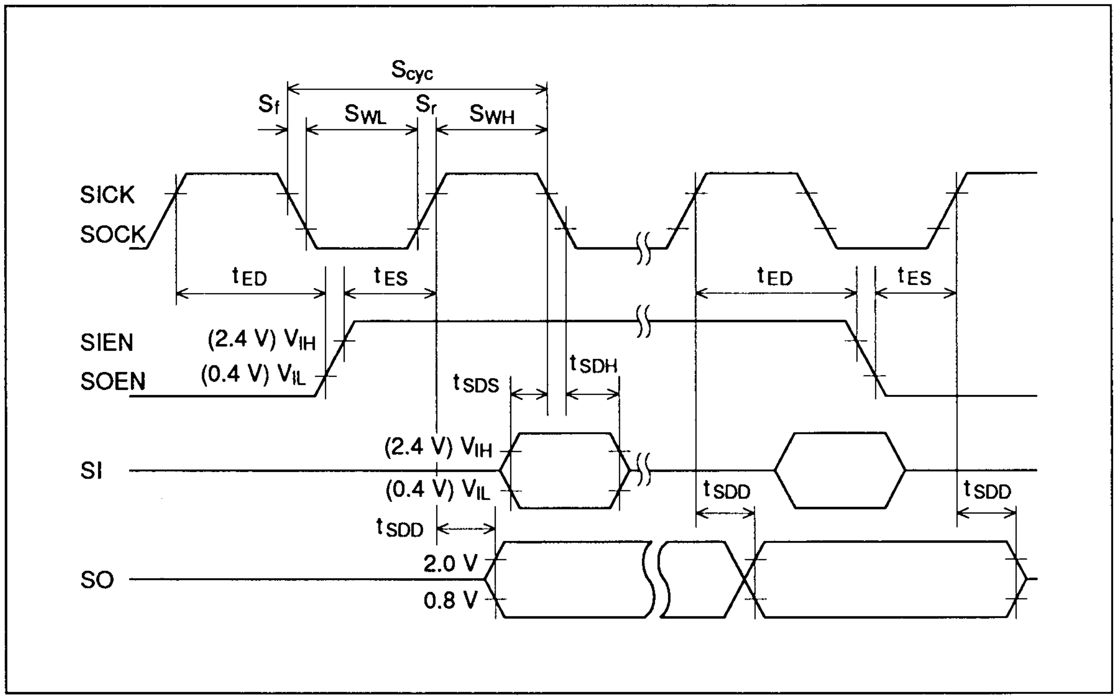


Figure 3 Serial I/O Timing

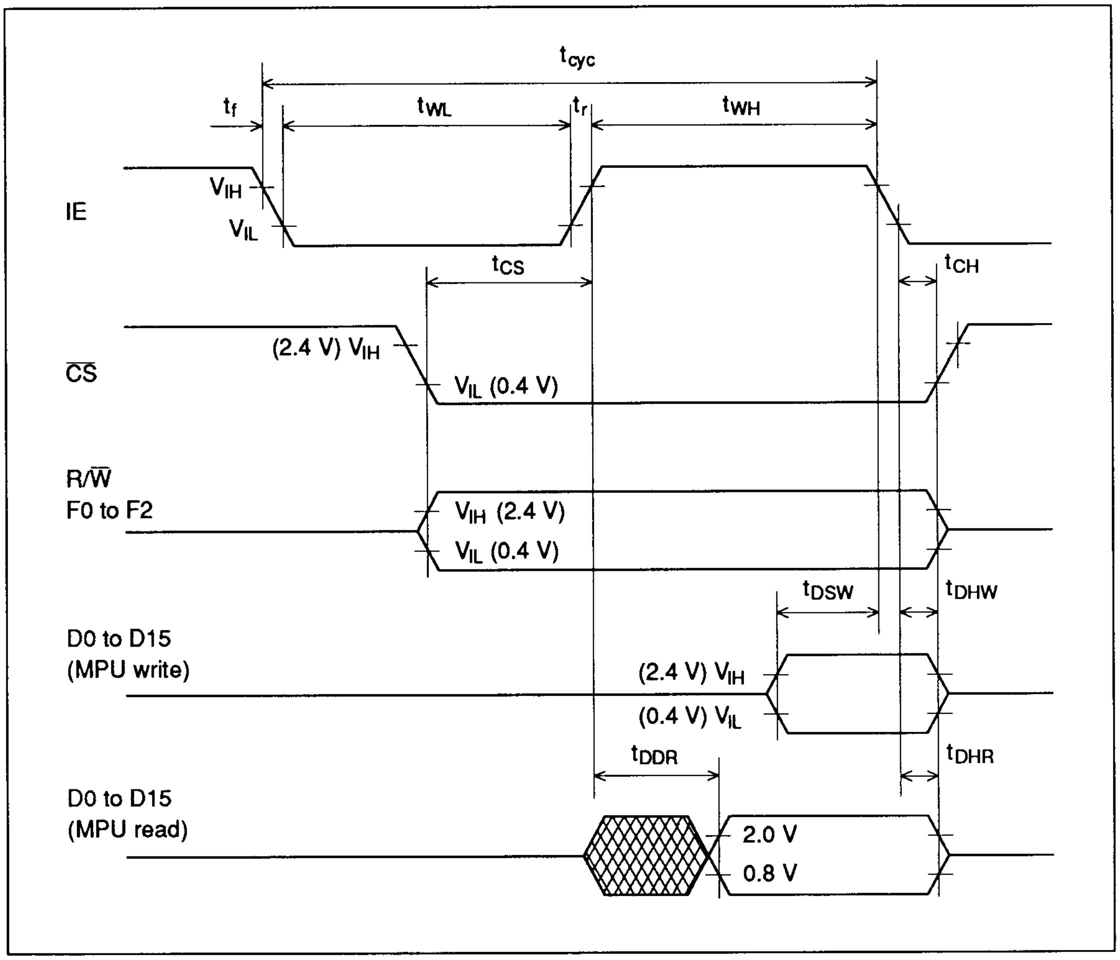


Figure 4 Parallel I/O Timing

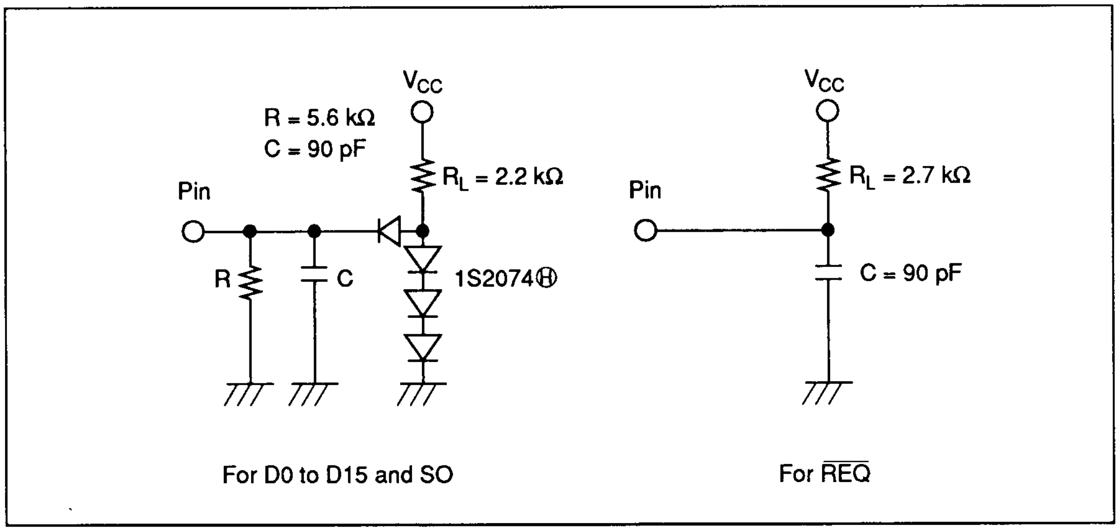


Figure 5 A.C. Testing Load Circuits

Signal Descriptions

Symbol	Pin No.	I/O	Description
			Power supply
V_{CC}	20		+5 V power supply
GND	1		Ground
			Clock
OSC	40	Input	Input clock: Apply a 32-MHz clock signal.
SCLK	39	Output	Internal system clock: Outputs a 8-MHz clock except during reset.
			Serial I/O
SICK	33	Input	Serial input clock: Serial data is input at the falling SICK edge.
SIEN	30	Input	Serial input enable: Serial data can be input when SIEN is high.
SI	29	Input	Serial input: Serial data must be input MSB first.
SOCK	31	Input	Serial output clock: Serial data is output after the rising SOCK edge.
SOEN	32	Input	Serial output enable: Serial data can be output when SOEN is high.
SO	36	Output	Serial output: Three state serial output. After SOEN goes low, SO enters a high-impedance state after the next rising SOCK edge.
			Parallel input
F0 to F2	24, 25, 27	Input	Parallel I/O Control: Used for specifying the transfer of commands or data.
$\overline{CS}$	23	Input	Chip select: While $\overline{CS}$ is low, parallel input/output (F0 to F2, R/$\overline{W}$, IE and D0 to D15) are enabled.
R/$\overline{W}$	22	Input	Read/write Control: Input for deciding the direction of transferring data, which can be output while R/$\overline{W}$ is high and input while R/$\overline{W}$ is low.
IE	21	Input	Data bus enable: Data can be input or output while IE is high.
			Parallel input/output data bus
D0 to D15	4 to 19	Input/Output	Data bus: Three state bus for transferring data to/from a microprocessor. Use mode select pins, MODE1 and 2 to select either 8- or 16-bit transfer. If 8-bit transfer is selected, ground D8 through D15.
			Auxiliary output
$\overline{REQ}$	34	Open drain output	Data transfer request: Notifies the microprocessor that data is ready for transfer by the level output (low level). $\overline{REQ}$ enters a high-impedance state after the transfer is completed.
			Operating mode select
MODE1	28	Input	Operating mode select pins: These can select following modes:
MODE2	3	Input	MODE1 low and MODE2 low selects serial coding mode.
			MODE1 low and MODE2 high selects serial decoding mode.
			MODE1 high and MODE2 low selects byte transferring mode.
			MODE1 high and MODE2 high selects word transferring mode.
LAW	35	Input	μ-/A-law select pin: μ-law is selected when LAW is low, and A law is selected LAW is high.
			Reset
$\overline{RES}$	38	Input	LSI reset input.
			Other pins
VL1, VL2	2, 26	Input	Voltage level 1 and 2: Test pins (connect to ground)
VH	37	Input	Voltage high: Test pin (connect to +5 V)

General Description

The HD81803 has two operating modes: Parallel mode is used to operate the HD81803 in combination with a microprocessor (Figure 6), and; serial mode is used for stand-alone operations (Figure 7). The MODE1 signal determines which mode is selected. Serial mode can be used for either input or output ADPCM serial data.

Physically, the HD81803 has two serial interfaces and one parallel interface. One serial interface is for the input and the other is for the output of either PCM or ADPCM data.

The serial interface input or output can be connected directly to a PCM CODEC. For serial PCM input, SI pin can be directly connected to a PCM out pin of a PCM CODEC, the SIEN enable signal is used to provide 8 kHz synchronization and the SICK clock signal is used to provide bit synchronization. For serial PCM output, SO pin is used can be directly connected to a PCM in pin of a PCM CODEC, the SOEN and SOCK signals are used.

The parallel interface allows a microprocessor to control the HD81803 and to send or receive speech data. Parallel I/O control signals F0 to F2 are used to indicate whether the data on the bus (D0 to D15) is command or speech data.

Commands can be transferred at any time while the HD81803 is in operation.

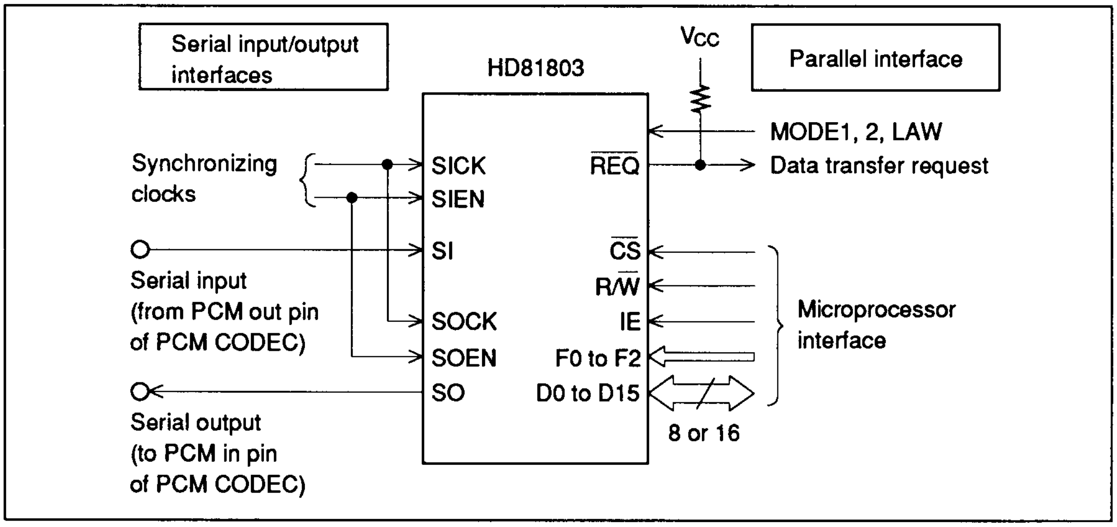


Figure 6 Example Operation in the Parallel Mode

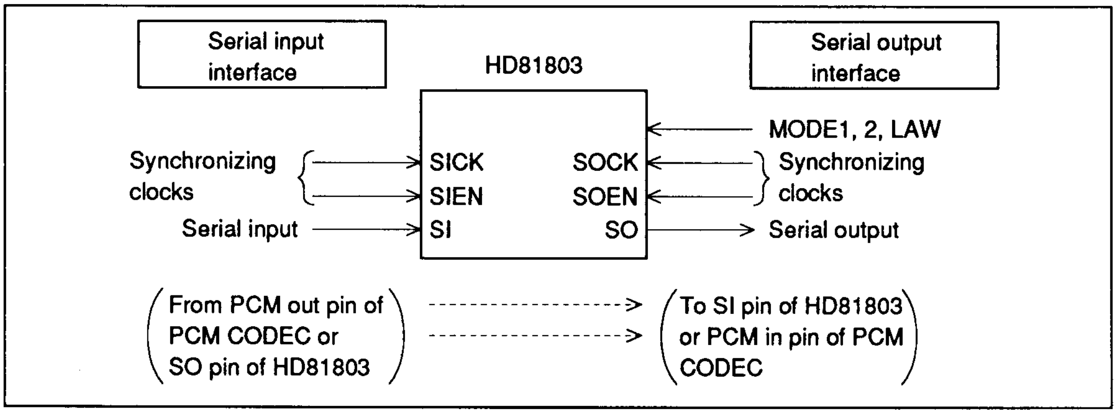
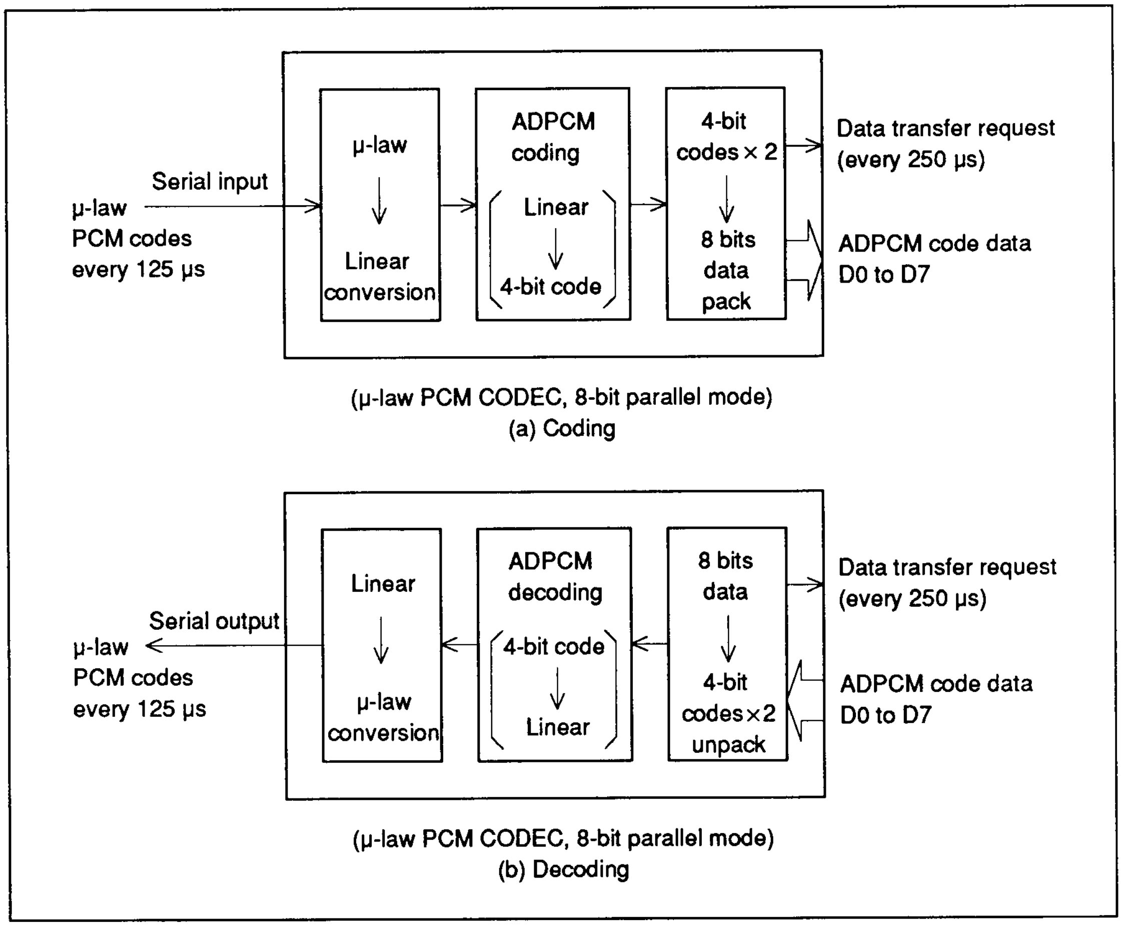


Figure 7 Example Operation in the Serial Mode

Figure 8 shows the data flow for the 8-bit parallel mode. The µ-law PCM data arriving at the serial input is converted to ADPCM format and output on the data bus (in this case, 8 bits at a time). The HD81803 compresses input speech data from 8-bit codes into 4-bit codes. Since the sampling frequency is 8 kHz, the speech data is compressed from 64 kbits/s to 32 kbits/s. Since the data bus can be 8 or 16 bits wide, one, two, or four samples of the coded data can be output at once.

For decoding, ADPCM data is input via the parallel interface and converted into µ-law PCM which is output from the serial output.



(µ-law PCM CODEC, 8-bit parallel mode)
(a) Coding

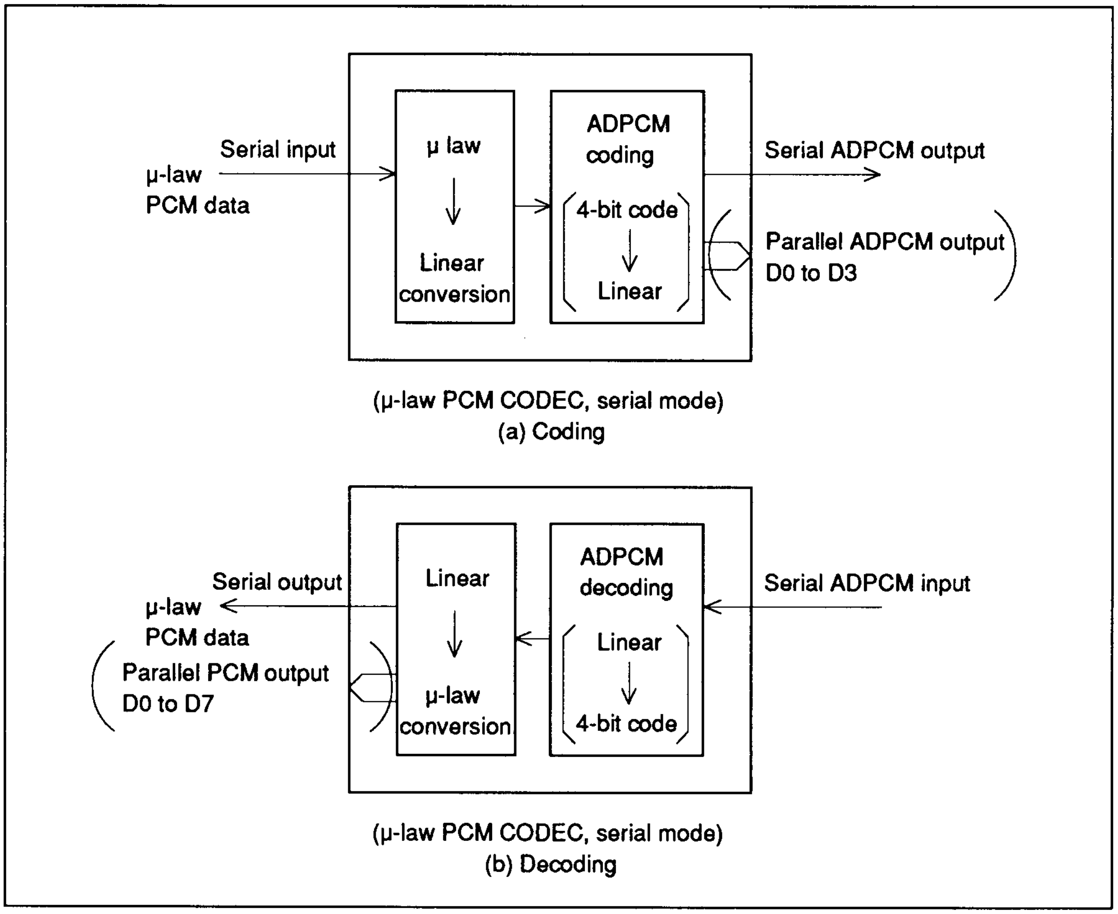
(µ-law PCM CODEC, 8-bit parallel mode)
(b) Decoding

Figure 8 Data Flow in the Parallel Mode

Figure 9 shows the data flow in the serial mode. For coding, μ-law PCM data arriving at the serial input is converted into ADPCM data and output from the serial output (See Figure 9a).

For decoding, ADPCM data arriving at the serial input is converted into μ-law PCM data and output from the serial output (See Figure 9b).

During coding (decoding), ADPCM (PCM) data can also be output from the parallel interface.



(μ-law PCM CODEC, serial mode)
(a) Coding

(μ-law PCM CODEC, serial mode)
(b) Decoding

Figure 9 Data Flow in the Serial Mode

Operating Mode Select Pins

The HD81803 has three operating mode select pins: MODE1, MODE2, and LAW (see Table 1). Command 0 (see "Parallel Interface") can also be used to select the operating mode.

Serial Interface

The serial interfaces allow the HD81803 to be connected directly to a PCM CODEC. The LAW signal specifies whether the PCM CODEC uses µ-law or A-law coding (see Table 2).

Figures 10 and 11 show example PCM CODEC interface connections for the parallel and serial modes, respectively. The PCM CODECs* used in these examples, which contain A/D and D/A converters with 12- or 13 bit precision, S/H circuitry, and filters all on a single chip, can effectively reduce the scale and cost of an analog I/O system. To synchronize the serial interfaces with the PCM CODEC, the serial I/O clock and enable signals are supplied externally.

In the serial mode, ADPCM data can be input and output in serial as shown in Figure 7. Since ADPCM codes are 4 bits long, the serial I/O enable pulse is set to be 4 bits wide (see Figure 12). Figure 13 shows an example of timing circuit.

Table 1 Operating Mode Selection Pins

MODE1 Pin	MODE2 Pin	LAW Pin
0: Serial mode	0: Coding mode	0: µ-law
	1: Decoding mode	1: A-law
1: Parallel mode	0: Byte transferring mode	0: µ-law
	1: Word transferring mode	1: A-law

In the serial mode following a reset, the HD81803 determines its status from the MODE1, 2 and LAW signals and starts processing.

In the parallel mode, the HD81803 will not start until a command is received.

Table 2 PCM CODEC Types for Serial Interfaces

Type	Code length	Sampling frequency	Remarks
A-law PCM CODEC	8 bits	8 kHz	LAW pin = 1
µ-law PCM CODEC	8 bits	8 kHz	LAW pin = 0

* 44237C (A-law), 44238C (µ-law), 44277P (A-law), 44278P (µ-law)

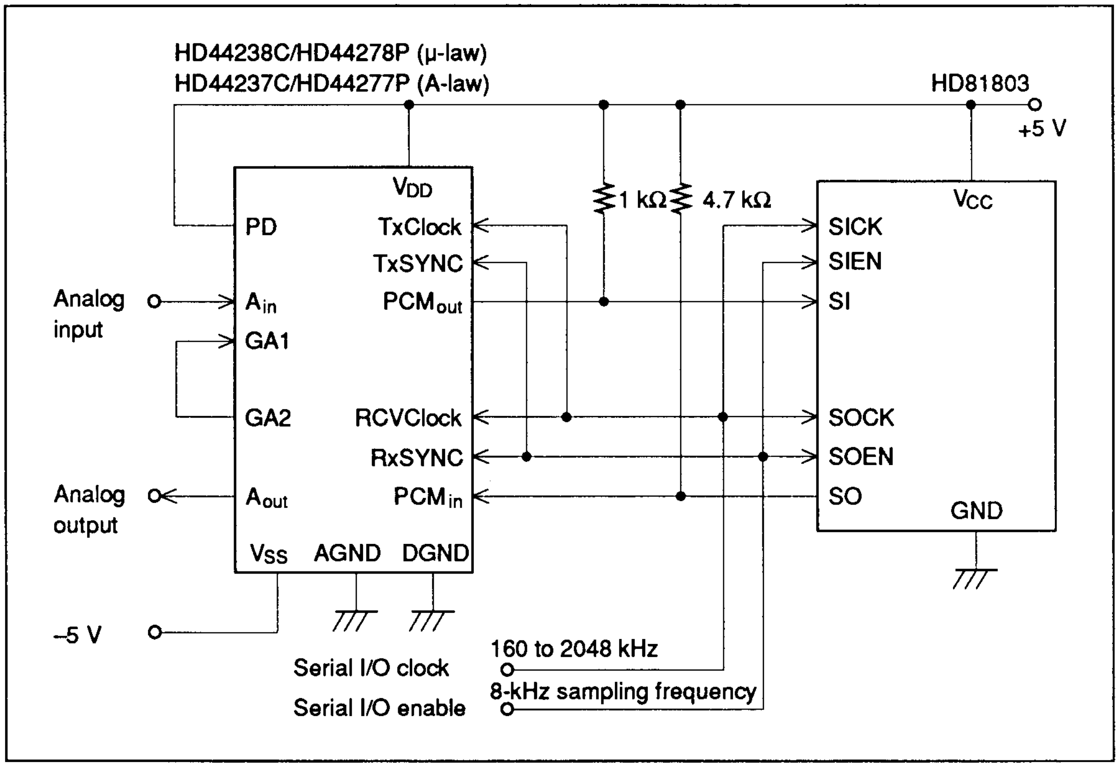


Figure 10 PCM CODEC Interface for the Parallel Mode

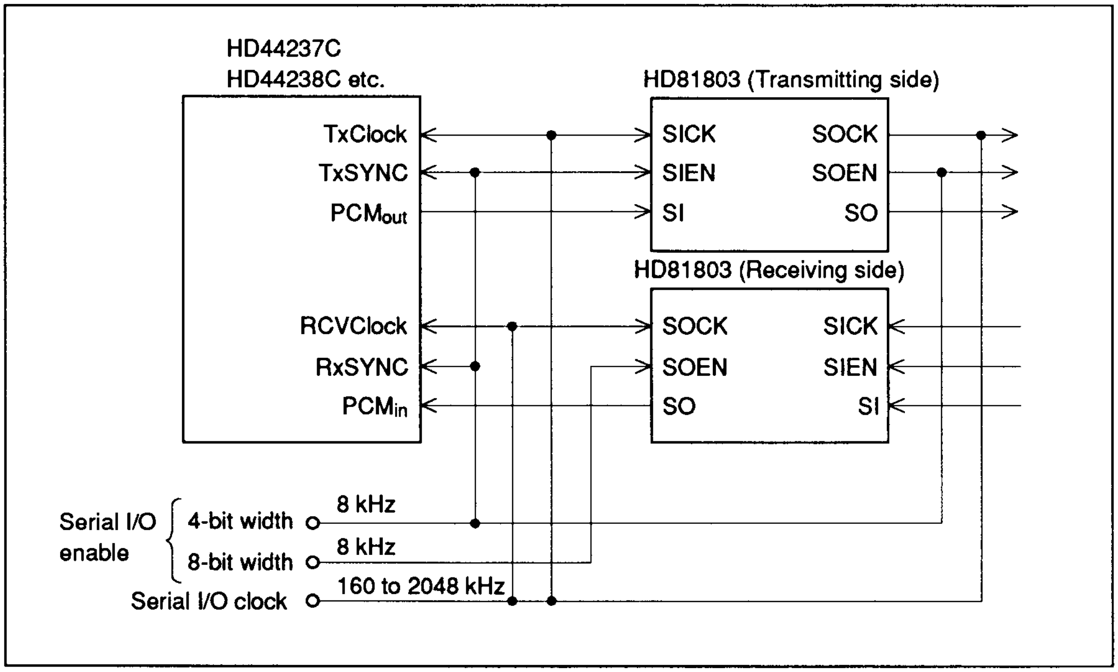


Figure 11 PCM CODEC Interface for the Serial Mode

Serial I/O clock

Serial I/O is performed by using synchronizing clocks (serial input clock SICK, serial output clock SOCK, serial input enable SIEN, and serial output enable SOEN) and input/output datas.

Figure 12 (a) shows the serial I/O timing for the PCM CODEC interface and Figure 12 (b) shows the timing for the ADPCM data interface. Figure 13 shows an example serial I/O timing circuit. In this example, serial I/O enable pulses SIEN and SOEN may be 4 bits or more in width, and only the enabled bits are output from SO. Thus, the output data might contain more bits than required (4 bits).

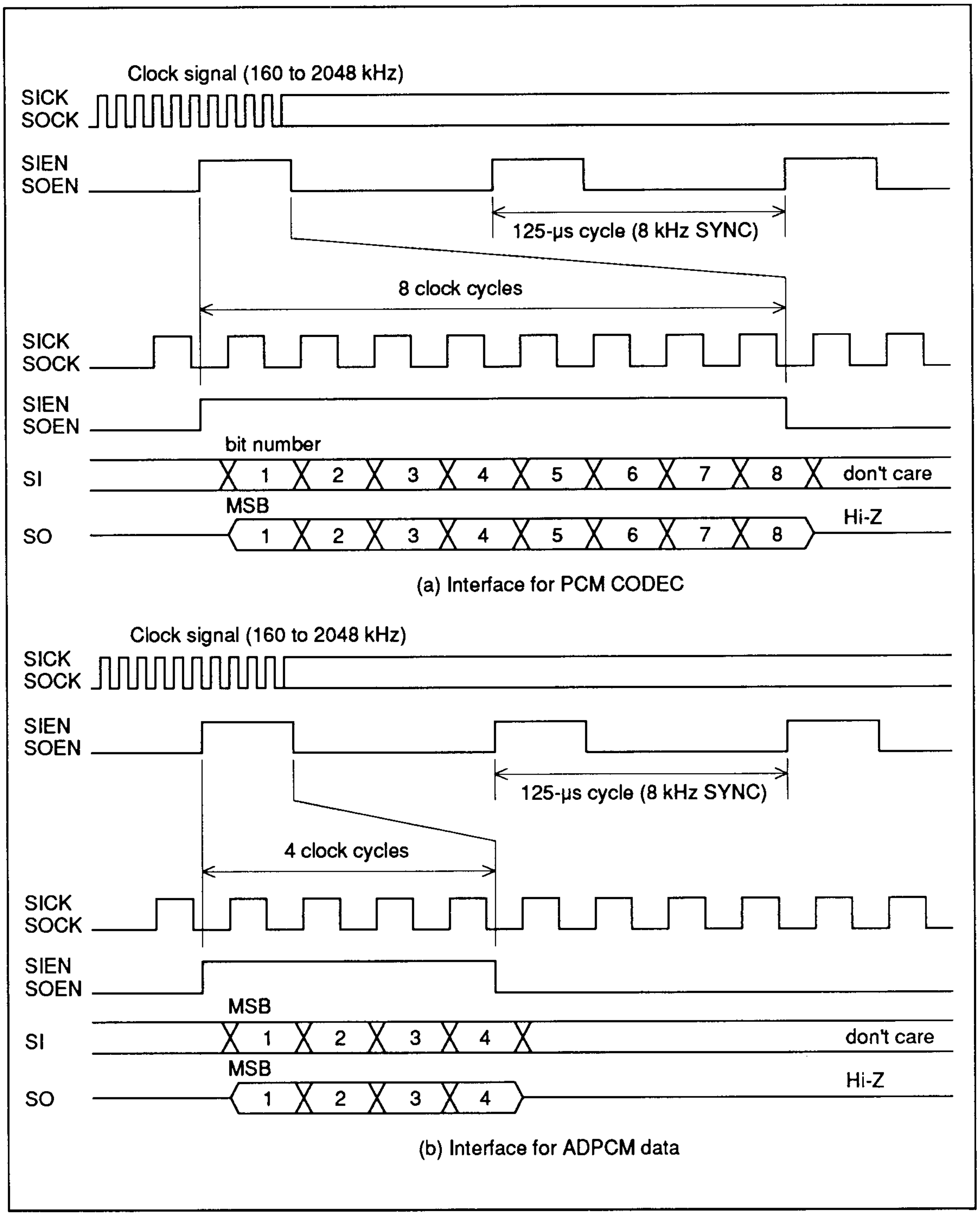


Figure 12 Serial Interface Timing

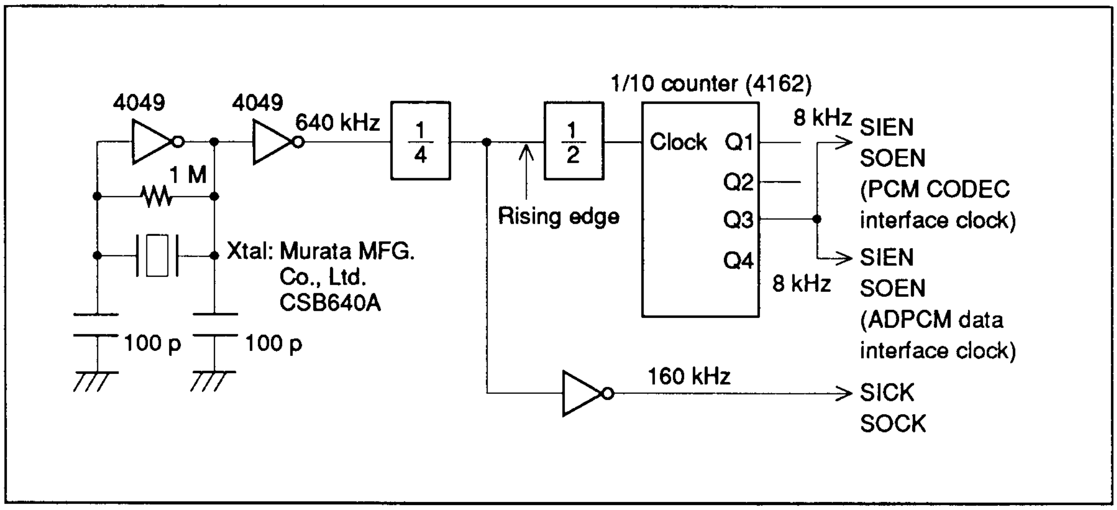


Figure 13 Sample Serial I/O Timing Circuit

Parallel Interface

Command and Data Transfer

In the parallel mode, a command transfer is needed to start the internal processing the HD81803. After the internal processing the HD81803 has been started, it can request the microprocessor to transfer data using $\overline{REQ}$.

A command transfer involves sending 16 bits, but since the high-order byte is fixed, only the lower eight bits are significant.

Upon receiving a data transfer request, the microprocessor does either a read or write. Data transfer requests are generated after a command other than an initialize or idle command is issued. The contents to be transferred, transfer direction, and transfer procedure are defined below for each command.

When an 8-bit data bus is used, ground pins D8 through D15.

Commands

(1) Command overview

After a reset, the HD81803 is initialized and waits for a command input. In this state, operations such as mode selection, coding, and decoding etc can be controlled by transferring a command from the microprocessor via the parallel interface. Table 3 lists the commands. (⟶ indicates parallel I/O and —> indicates serial I/O.)

Table 3 HD81803 commands

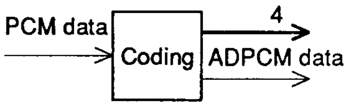
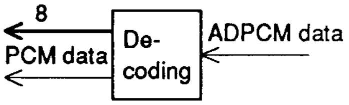
Pin level					
F	**$R/\overline{W}$**	**Command**	**Code*1**	**Description**	**Data flow**
*2 F0 = L/H F1 = L F2 = H	$R/\overline{W}$ = L	Initialize command	$0000	Sets internal MODE1, MODE2, and LAW values (equivalent to hardware reset).	
		Idle command	$0001	Places the HD81803 in the command waiting state. This command is used to stop the processing of other commands.	
		Serial coding start command	$0002	After this command is transferred, PCM data is input and coded (ADPCM) data is output.	
		Serial decoding start command	$0003	After this command is transferred, ADPCM data is input and decoded (PCM) data is output.	

Table 3 HD81803 commands (cont)

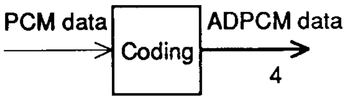
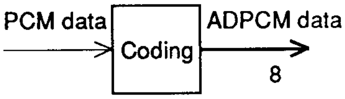
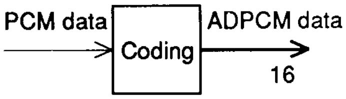
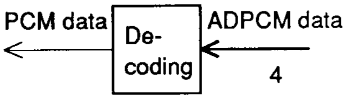
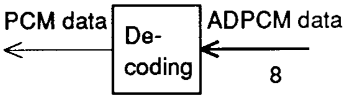
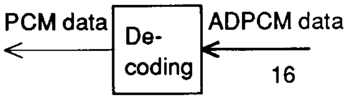
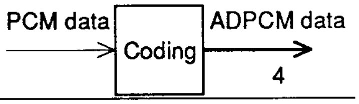
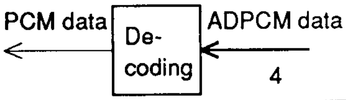
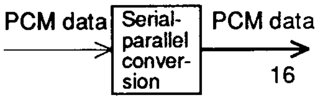
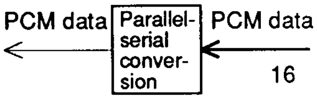
Pin level					
F	R/W̄	Command	Code*1	Description	Data flow
F0 = L/H F1 = L F2 = H	R/W̄ = L	Parallel coding start command 1	$0004	After this command is transferred, coded (ADPCM) data will begin to be output.	
		Parallel coding start command 2	$0005	The commands 1, 2, 3 differ according to the number of data bus bits (4, 8, or 16) used to carry coded (ADPCM) data.	
		Parallel coding start command 3	$0006		
		Parallel decoding start command 1	$0007	After this command is transferred, coded (PCM) data is output when ADPCM data is input.	
		Parallel decoding start command 2	$0008	The commands 1, 2, 3 differ according to the number of data bus bits (4, 8, or 16) used to carry ADPCM data.	
		Parallel decoding start command 3	$0009		
			$000A	Reserved.	
			$000B	Reserved.	
		Coding test start command	$000C	After this command is transferred, the coding test starts.	
		Decoding test start command	$000D	After this command is transferred, the decoding test starts.	

Table 3 HD81803 commands (cont)

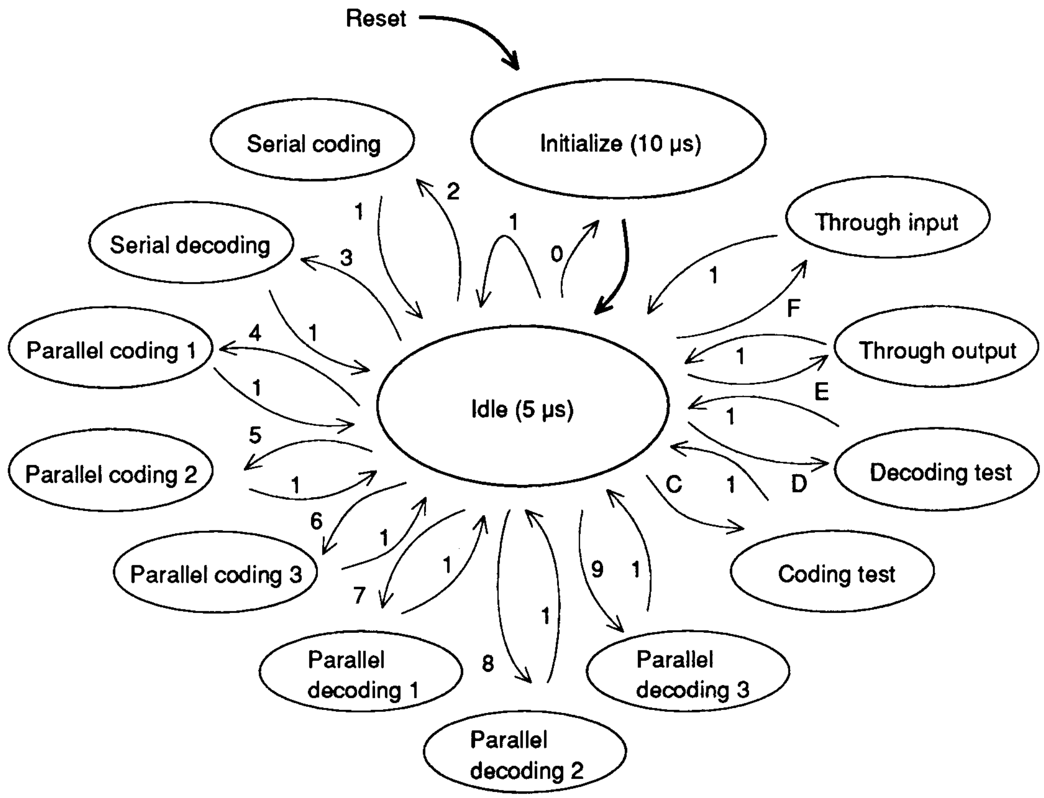
Pin level					
F	$R/\overline{W}$	Command	Code[*1]	Description	Data flow
F0 = L/H F1 = L F2 = H	$R/\overline{W}$ = L	Through output start command	$000E	After this command is transferred, PCM data is input in serial and output in parallel.	
		Through input start command	$000F	After this command is transferred, PCM data is input in parallel and output in serial.	

*1. This shows the value of data transferred on the data bus (D15–D0). In the word transferring mode, data is transferred twice using D0 through D7 (starting with the upper byte).

*2. For the byte transferring mode, leave F0 high. In the 8-bit parallel mode, set F0 low to send the high-order byte of the command, and high to send the low-order byte.

Example: Transferring a parallel coding start command
Word transfer: Send D15 through D0 with $0005 and F2, F1, F0 = 101.
Byte transfer: Send D7 through D0 with $00 and F2, F1, F0 = 100, then D7 through D0 with $05 and F2, F1, F0 = 101.

(2) State transition diagram



- The hexadecimal numbers in the diagram correspond to the four low-order bits of the transfer command.

0. Initialize command
1. Idle command
2. Serial coding start command
3. Serial decoding start command
4. Parallel coding start command 1
5. Parallel coding start command 2
6. Parallel coding start command 3
7. Parallel decoding start command 1
8. Parallel decoding start command 2
9. Parallel decoding start command 3
C. Coding test start command (for digital sequence tests)
D. Decoding test start command (for digital sequence tests)
E. Through output start command
F. Through input start command

- A bold line indicates that the state transition occurs automatically (i.e. the idle state is entered automatically after initialization).
- The idle command can be issued to stop command execution.
- For.the initialize and idle states, the value in parenthesis indicates the execution time. During this period other commands cannot be executed.

(3) Functional description of commands

(a) Initialize command

The initialize command initializes the internal state of the HD81803, in the same way as a hardware reset. When this command is transferred via the parallel interface, the initializing of the internal state of the HD81803 will begin according to the mode select inputs MODE1, MODE2, and LAW. After 10 µs the HD81803 enters the idle state.

(b) Idle command

The idle command terminates the execution of other commands, and places the HD81803 in the idle state.

(c) Serial coding start command

After this command is transferred, the HD81803 converts the PCM data received via the serial input into ADPCM data. The ADPCM data is output from the serial output, and can be also read via the parallel interface after the data transfer request signal $\overline{\text{REQ}}$ goes low. $\overline{\text{REQ}}$ goes low for every serial input sample (i.e. every 125 µs).

To stop the execution of this command, issue the idle command.

(d) Serial decoding start command

After this command is transferred, the HD81803 converts the ADPCM data received via the serial input into PCM data. The PCM data is output from the serial output, and can be also read via the parallel interface after the data transfer request signal $\overline{\text{REQ}}$ goes low. $\overline{\text{REQ}}$ goes low for every serial output sample (i.e. every 125 µs).

To stop the execution of this command, issue the idle command.

(e) Parallel coding start commands 1, 2, and 3

After this command is transferred, the HD81803 converts the PCM data received via the serial input into ADPCM data. The ADPCM data is output from the parallel output.

The ADPCM data can be read by the microprocessor after the data transfer request signal $\overline{\text{REQ}}$ goes low. $\overline{\text{REQ}}$ goes low: for every serial input sample when parallel coding start command 1 is issued; for every two input samples when parallel coding start command 2 is issued; and for every four input samples when parallel coding start command 3 is issued. Thus, for an 8-kHz sampling rate, read cycles would be needed every 125, 250 and 500 µs, respectively.

To stop the execution of this command, issue the idle command.

(f) Parallel decoding start commands 1, 2, and 3

After this command is transferred, the HD81803 converts the ADPCM data received via the parallel interface into PCM data. The PCM data is output from the serial output.

The ADPCM data can be sent by the microprocessor after the data transfer request signal $\overline{\text{REQ}}$ goes low. $\overline{\text{REQ}}$ goes low: for every serial output sample when parallel decoding start command 1 is issued; for every two output samples when parallel decoding start command 2 is issued; and for every four output samples when parallel decoding start command 3 is issued. Thus, for an 8-kHz sampling rate, write cycles would be needed every 125, 250 and 500 µs, respectively.

To stop the execution of this command, issue the idle command.

(g) Coding test start command

The coding test start command is used to check the ADPCM coding process of the HD81803 (digital sequence test defined in CCITT Recommendation G.721) via the parallel interface. After this command is transferred, the HD81803 receives the PCM data via the parallel interface, converts it into ADPCM data, and outputs the ADPCM data back via the parallel interface.

The PCM data can be written and the ADPCM data can be read after the data transfer request signal REQ goes low. There is no limit on the response time. After this command is transferred, the HD81803 requests to write PCM data first. A request to write PCM data is always followed by a request to read ADPCM data. Thus, write and read requests are issued in sequence.

To stop the execution of this command, issue the idle command.

(h) Decoding test start command

The decoding test start command is used to check the ADPCM decoding process of the HD81803 (digital sequence tests defined in CCITT Recommendation G.721) via the parallel interface. After this command is transferred, the HD81803 receives the ADPCM data via the parallel interface, converts it into PCM data, and outputs the PCM data back via the parallel interface.

The ADPCM data can be written and the PCM data can be read after the data transfer request signal $\overline{\text{REQ}}$ goes low. There is no limit on the response time. After this command is transferred, the HD81803 requests to write ADPCM data first. A request to write ADPCM data is always followed by a request to read PCM data. Thus, write and read requests are issued in sequence.

To stop the execution of this command, issue the idle command.

(i) Through output start command

The through output start command causes the HD81803 to convert the PCM data received via the serial input into parallel. The parallel PCM data is output from the parallel output.

The parallel PCM data can be read after the data transfer request signal $\overline{\text{REQ}}$ goes low. One read request is needed for every two serial input samples. Thus, for an 8-kHz sampling rate, read cycles would be needed every 250 µs.

To stop the execution of this command, issue the idle command.

(j) Through input start command

The through input start command causes the HD81803 to convert the PCM data received via the parallel interface into serial. The serial PCM data is output from the serial output.

The parallel PCM data can be written after the data transfer request signal $\overline{\text{REQ}}$ goes low. One write request is needed for every two serial output samples. Thus, for an 8-kHz sampling rate, write cycles would be needed every 250 µs.

To stop the execution of this command, issue the idle command.

Data transfer

(1) Input/output data format

As explained in (3) above, the width of the data bus used in the execution of a command depends on the command transferred. Details are given below.

(a) ADPCM data

ADPCM coded data is transferred to/from the microprocessor 4, 8, or 16 bits at a time via the parallel interface. One ADPCM code consists of 4 bits. Figure 14 shows the format for transferring ADPCM codes. (n indicates the sequence in which the codes are transferred.)

(b) PCM data

PCM data is transferred to/from the microprocessor as 8 or 16 bits at a time via the parallel interface. One PCM code consists of 8 bits. Figure 15 shows the format for transferring PCM codes.

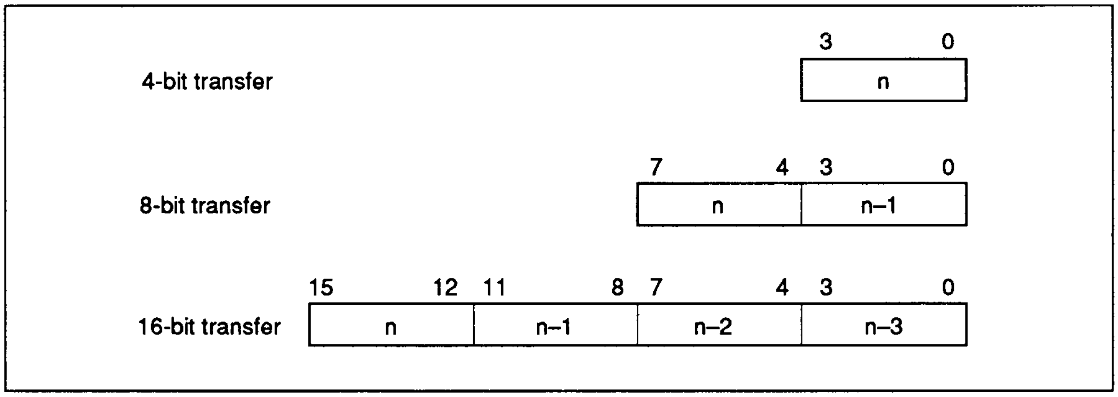


Figure 14 Parallel I/O Format for ADPCM Data

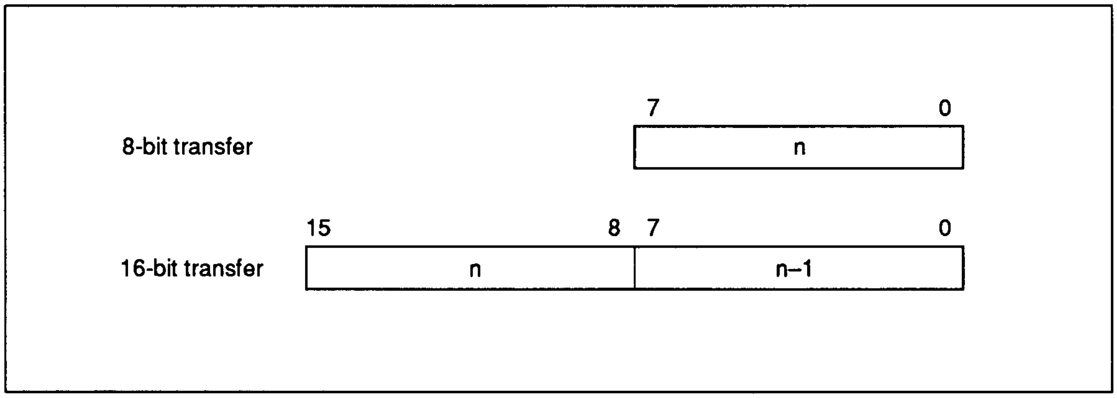


Figure 15 Parallel I/O Format for PCM Data (Through Output/Input Start Commands)

(2) Data input/output operations

Table 4 shows the data input/output operations for the parallel interface.

Table 4 HD81803 Parallel I/O Operations

Pin level			Data bus format	
F	R/$\overline{W}$	Operation	D15 ◄——► D0	Description

F0 = L
F1 = H
F2 = L
	R/$\overline{W}$ = H	DATA-U READ
		Upper data read
			15 8 7 0
			Undefined DATA
				Reads only the high-order byte of 16-bit data when an 8-bit bus is used.

F0 = H
F1 = H
F2 = L
		DATA-L READ
		Lower data read
			(1) 15 4 3 0
			Undefined DATA
				Reads only the four low-order bits when an 8-bit bus is used.

			(2) 15 8 7 0 Undefined DATA	Reads 8 bits or the low-order byte of 16-bit data when an 8-bit bus is used.
		DATA-W READ Word data read	15 0 DATA	Reads all 16 bits when a 16-bit bus is used.

F0 = L
F1 = H
F2 = L
	R/$\overline{W}$ = L	DATA-U WRITE
		Upper data write
			15 8 7 0
			don't care DATA
				Writes the high-order byte of 16-bit data when an 8-bit bus is used.

F0 = H
F1 = H
F2 = L
		DATA-L WRITE
		Lower data write
			(1) 15 4 3 0
			don't care DATA
				Writes the four low-order bits when an 8-bit bus is used.

			(2) 15 8 7 0 don't care DATA	Writes 8 bits or the low-order byte of 16-bit data when an 8-bit bus is used.
		DATA-W WRITE Word data write	15 0 DATA	Writes all 16 bits when a 16-bit bus is used.

Examples

1. Parallel coding start command 1 uses DATA-L READ.
2. Parallel decoding start command 3 uses DATA-W WRITE, or DATA-U WRITE and DATA-L WRITE.

(3) Data transfer request ($\overline{\text{REQ}}$)

(a) $\overline{\text{REQ}}$ is an open-drain output pin which requires an external pull-up resistor.

(b) $\overline{\text{REQ}}$ is used to request the microprocessor to transfer data to/from the HD81803 by the level output. When $\overline{\text{REQ}}$ goes low, the microprocessor must read or write data (except when transferring a serial coding/decoding start command).

The timing between $\overline{\text{REQ}}$ and the microprocessor's response are given below. The microprocessor must transfer data within 100, 200, or 400 µs after $\overline{\text{REQ}}$ goes low.

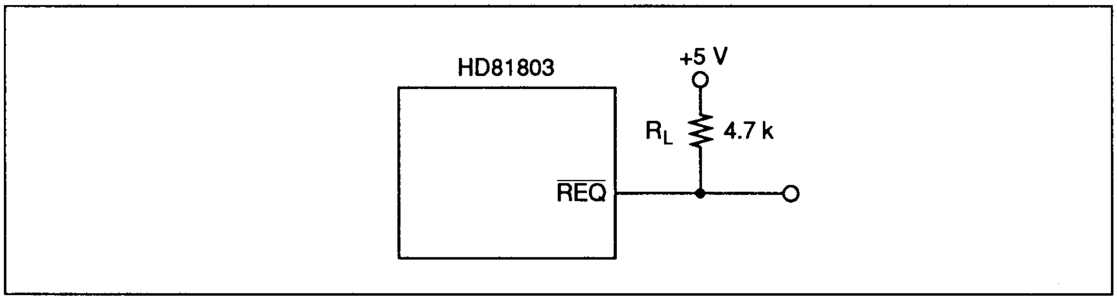


Figure 16 Data Transfer Request Output

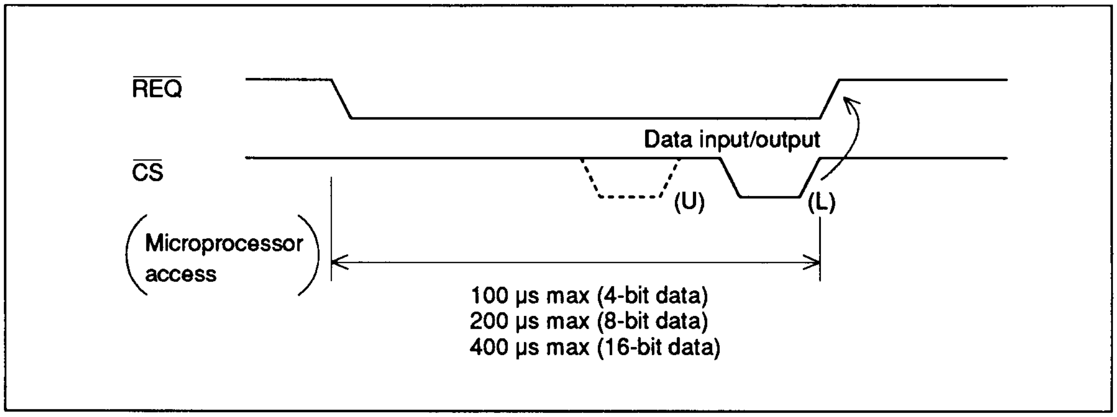


Figure 17 Maximum Microprocessor Response Time for $\overline{\text{REQ}}$ When Coding/Decoding (32-MHz Clock, 8-kHz Sampling Rate)

(4) Interfaces for 6800 and Z-80 series microprocessors

The HD81803 can be connected directly to 6800 series microprocessors by the parallel interface without the need for extra hardware. For example, Figure 18 shows how the Hitachi 8-bit HD6303 microprocessor could be interfaced.

Since the HD81803 does not always require the IE clock, only minimal hardware is required to interface it to a Z-80 series microprocessor. (see Figure 19)

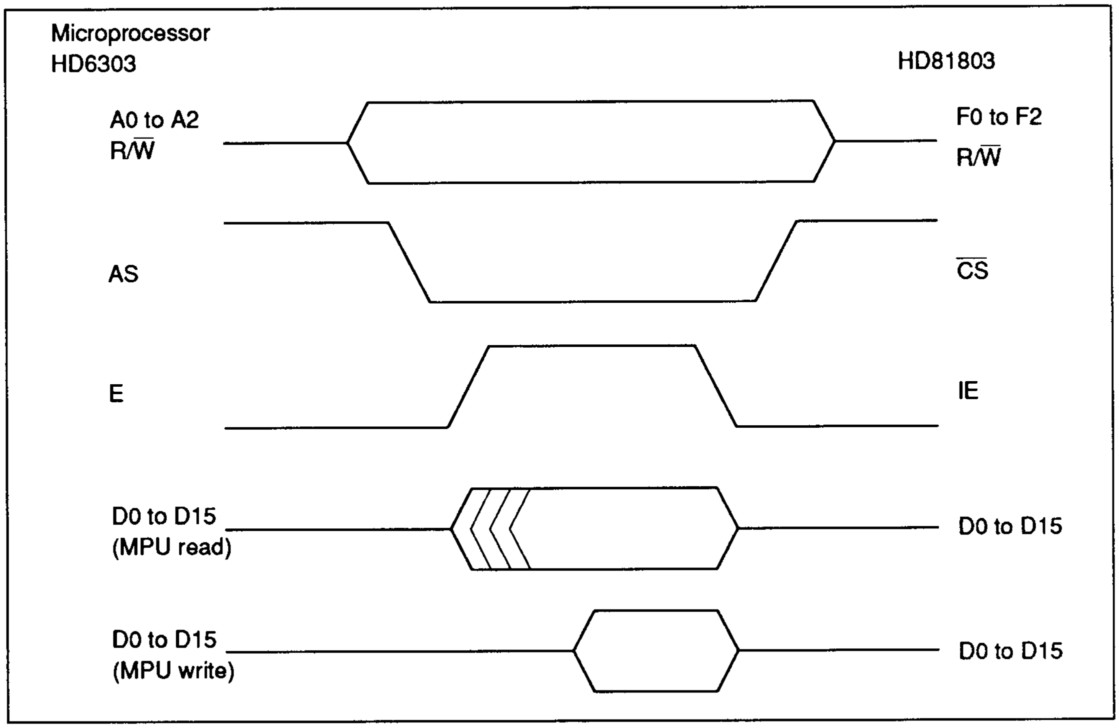


Figure 18 HD6303 Interface Signals

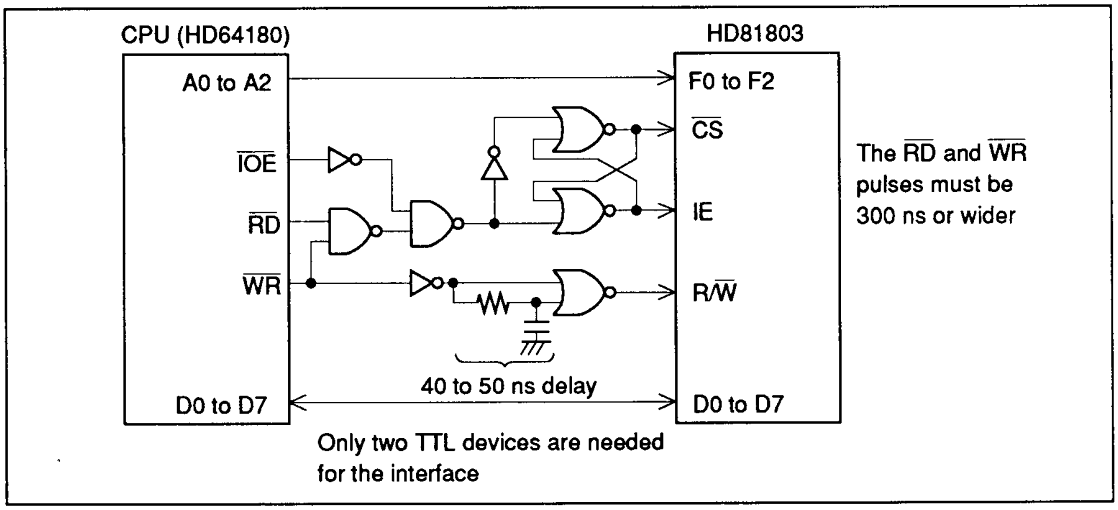

CPU (HD64180)
HD81803
A0 to A2
IOE
RD
WR
D0 to D7
F0 to F2
CS
IE
R/W
D0 to D7
40 to 50 ns delay
The RD and WR pulses must be 300 ns or wider
Only two TTL devices are needed for the interface

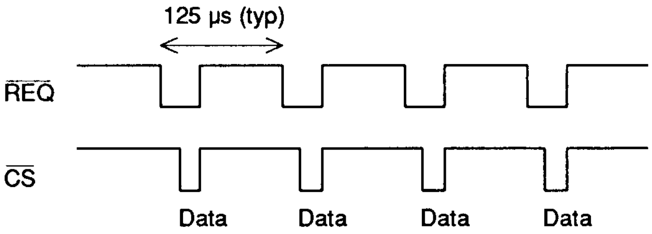
Figure 19 Example Interface for Z-80 Series Microprocessors

(5) Example response of a microprocessor to an ADPCM data transfer request (REQ)

When REQ goes low, the microprocessor must input or output data. Examples of the I/O timing on each states are given below.

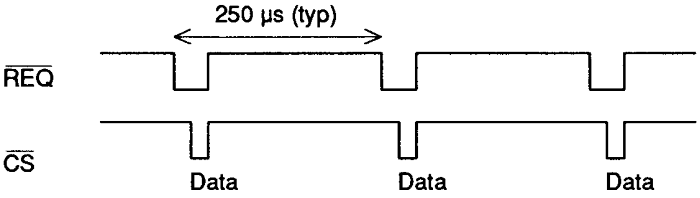
Examples

1. 4-bit data input/output in the 8-bit parallel mode



Applicable states: Serial coding
Serial decoding
Parallel coding 1
Parallel decoding 1

2. 8-bit data input/output in the 8-bit parallel mode



Applicable states: Parallel coding 2
Parallel decoding 2

— When transferring 16 bits of data, two 250-μs cycles are needed for an 8-bit bus, or one 500-μs cycle for a 16-bit bus.

Notes

The Phase relationship between serial input enable and serial output enable

The serial input enable (SIEN) and serial output enable (SOEN) signals should be basically the same. However, if a phase difference between them is unavoidable due to the system design, timing relationships should be as shown in Figure 20. The output from SO should be terminated before the input from SI is completed. The input from SI is completed at the rising edge 16 clock pulses from the rising edge of SIEN.

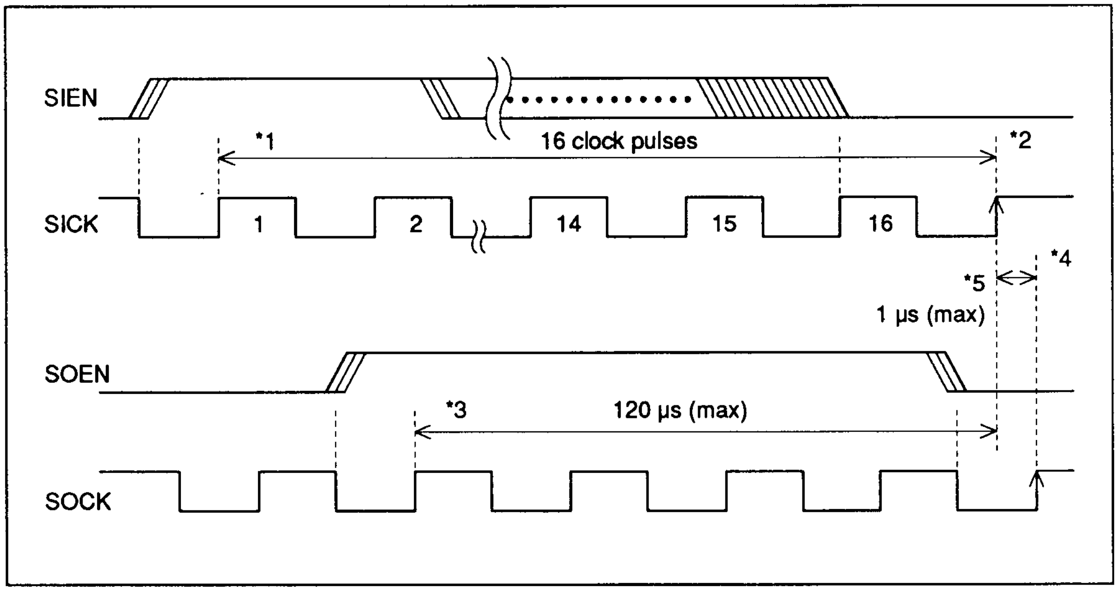


Figure 20 Timing Relation between Serial Input and Output Enable

*1: The first rising SICK edge after SIEN goes high
*2: The rising SICK edge 16 clock pulses later (The rising SIEN edge is ignored.)
*3: The rising SOCK edge after SOEN goes high
*4: The rising SOCK edge after SOEN goes low
*5: The minimum is undefined. That is, as in the usual case, this value might also be negative (*4 precedes *2).

Sample System Configuration

This section shows an example record/playback system consisting of the HD81803 and a host CPU HD6301 (Figure 21).

This design uses the HM62256 (256-kbit RAM) as a speech data RAM to record and playback speech data for 7–8 seconds. The microphone and power amplifiers are not shown in this figure.

Specifications

- Speech data memory: HM62256 (also used as system memory)
- CODEC: HD44238 μ-law CODEC
- Sampling frequency: 8 kHz
- Number of coding bits: 4

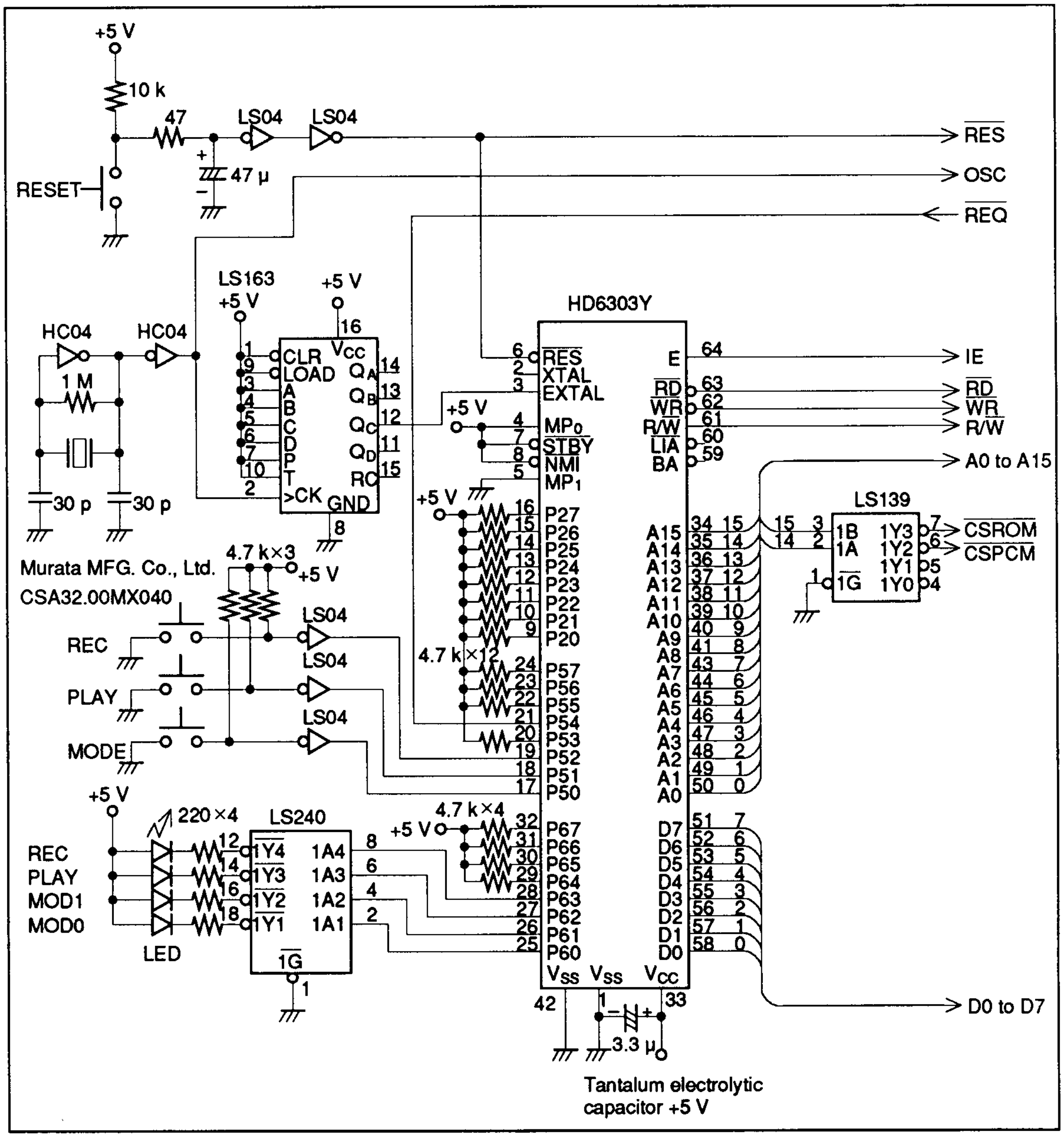
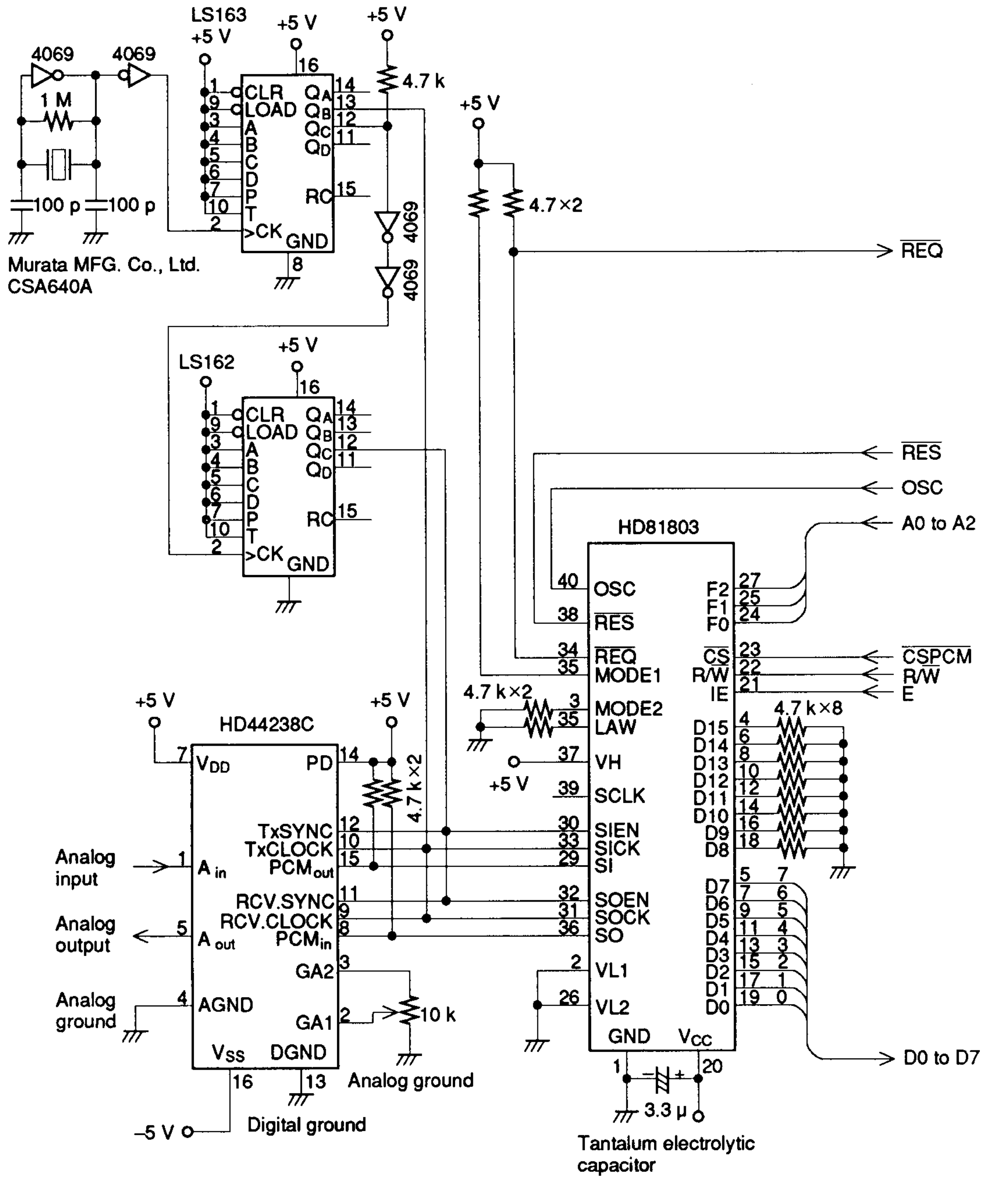


Figure 21 Example System Configuration (Microprocessor Section)



Separate the analog ground and digital grounds. Connect the analog signal lines (A_{in}, A_{out}, GA1, and GA2) as far from the digital signal lines as possible, and don't allow them to cross. Provide a bypass capacitor (3-47 μF) at the shortest connection between V_{CC} and ground.

Figure 22 Example System Configuration (ADPCM Section)

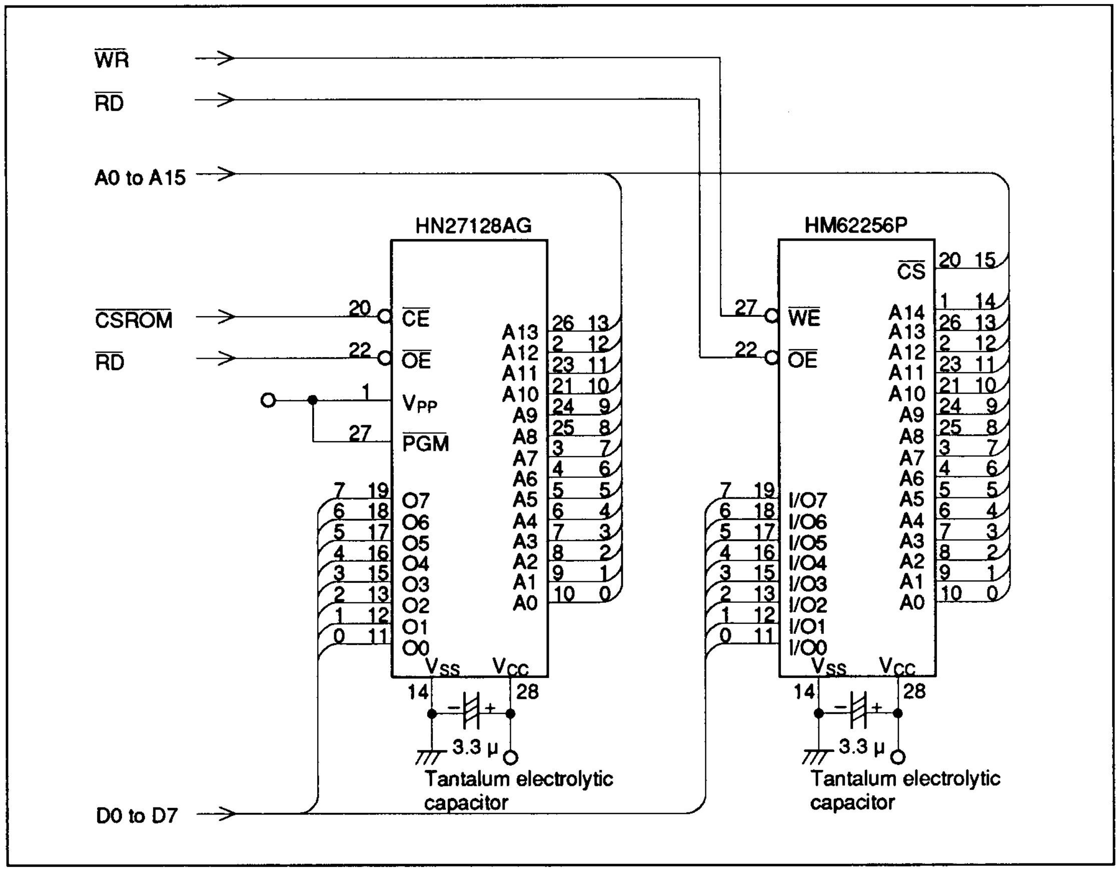


Figure 23 Example System Configuration (Memory Section)

Appendix (Terminology)

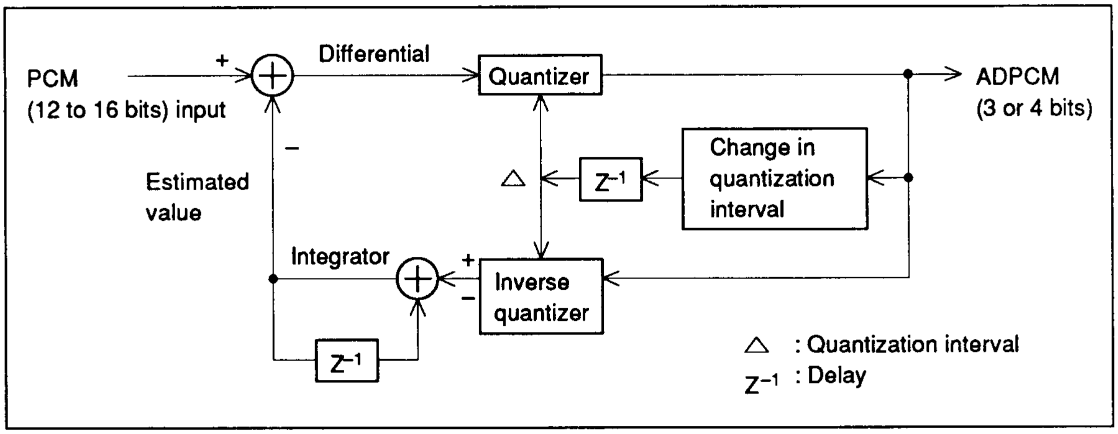
ADPCM

PCM (pulse code modulation) is a process for converting analog signals into digital signals. This process is employed in CD players and so on which can provide high-quality recording and playback of speech data with enough number of bits.

The ADPCM (adaptive differential pulse code modulation) is the one of methods which requires less data than PCM. In the ADPCM process, each differential between two adjacent samples is represented by the small number of bits (as PCM), while the quantization intervals (bit weight) of the PCM signals are made adaptive according to the differentials. Thus, high-quality speech data is available in a wide dynamic range thanks to ADPCM.

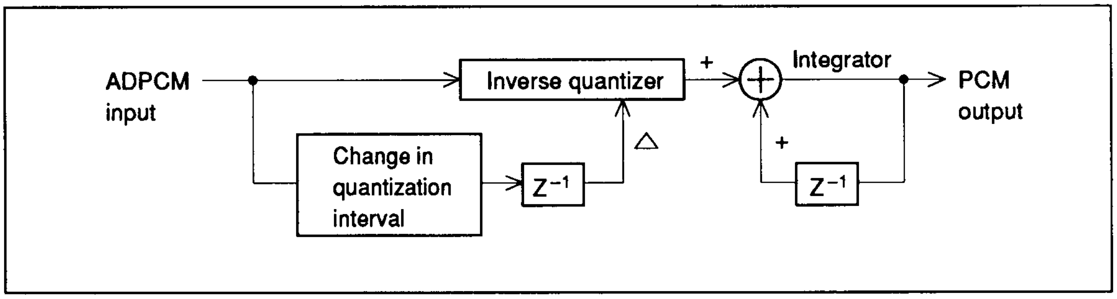
ADPCM Encoder

The ADPCM encoder quantizes the differential between the estimated and input values according to the quantization interval Δ.



ADPCM Decoder

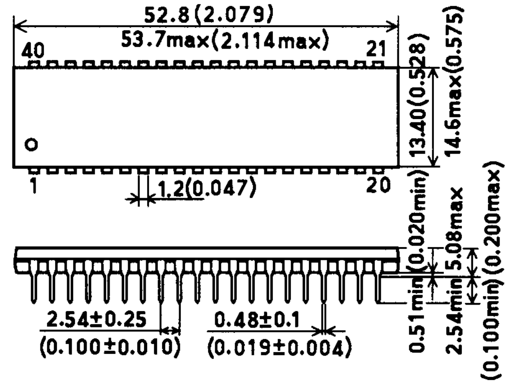
The ADPCM decoder, having the same configuration as the estimator in the ADPCM encoder above, reverses the coding procedure.

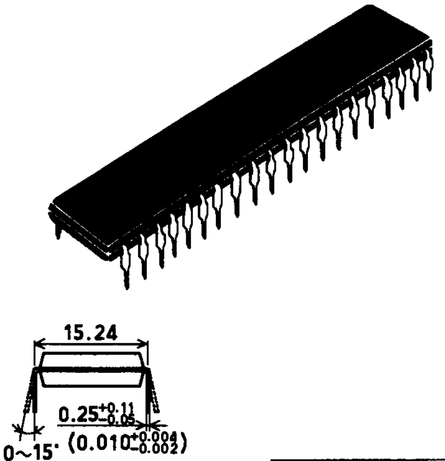


Reference: K. Nakata "Onsei" J. Acoustic Society of Japan, 1977, Corona Pub. Co., Ltd, Japan

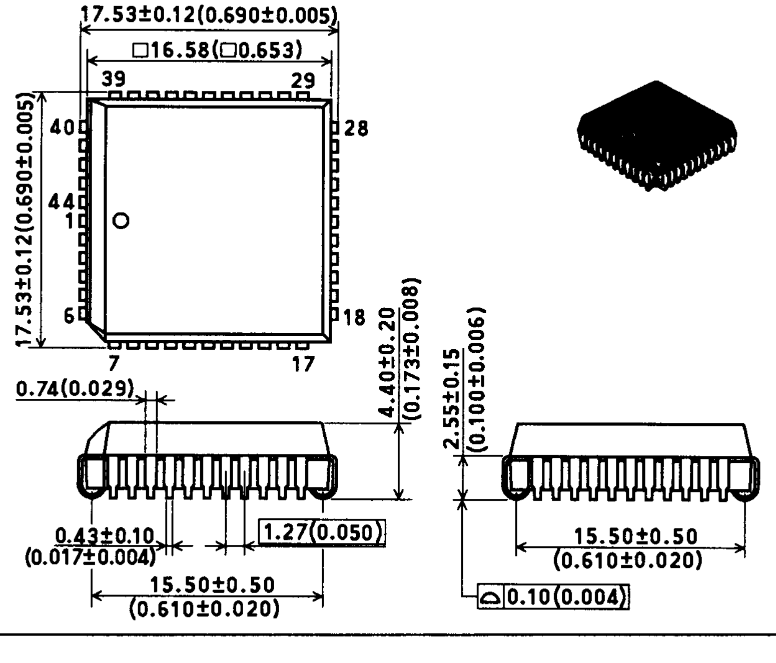
Package Dimensions

Unit: mm (inch)





Hitachi Code	DP-40
EIAJ	—
JEDEC	—



Hitachi Code	CP-44
EIAJ	—
JEDEC	—

When using this document, keep the following in mind:

1. This document may, wholly or partially, be subject to change without notice.
2. All rights are reserved: No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without Hitachi's permission.
3. Hitachi will not be held responsible for any damage to the user that may result from accidents or any other reasons during operation of the user's unit according to this document.
4. Circuitry and other examples described herein are meant merely to indicate the characteristics and performance of Hitachi's semiconductor products. Hitachi assumes no responsibility for any intellectual property claims or other problems that may result from applications based on the examples described herein.
5. No license is granted by implication or otherwise under any patents or other rights of any third party or Hitachi, Ltd.
6. MEDICAL APPLICATIONS: Hitachi's products are not authorized for use in MEDICAL APPLICATIONS without the written consent of the appropriate officer of Hitachi's sales company. Such use includes, but is not limited to, use in life support systems. Buyers of Hitachi's products are requested to notify the relevant Hitachi sales offices when planning to use the products in MEDICAL APPLICATIONS.

HITACHI

Hitachi, Ltd.
Semiconductor & IC Div.
Karukozaka MN Bldg., 2-1, Ageha-cho, Shinjuku-ku Tokyo 162, Japan
Tel: Tokyo (03) 266-9359
Fax: (03) 235-2375

For further information write to:

Hitachi America, Ltd.
Semiconductor & IC Div.
2000 Sierra Point Parkway
Brisbane, CA. 94005-1819,
U S A
Tel: 415-589-8300
Fax: 415-583-4207

Hitachi Europe GmbH
Electronic Components Div.
Central Europe Headquarters
Hans-Pinsel-Straβe 10A,
8013 Haar bei München,
West Germany
Tel: 089-46140
Fax: 089-463068

Hitachi Europe Ltd.
Electronic Components Div.
Northem Europe Headquarters
21 Upton Road,
Watford, Herts WD1 7TB,
United Kingdom
Tel: 0923-246488
Fax: 0923-224422

Hitachi Asia (Hong Kong) Ltd.
Unit 706, North Tower,
World Finance Centre,
Harbour City, Canton Road,
Tsimshatsui, Kowloon,
Hong Kong
Tel: 852-3-7359218
Fax: 852-3-7306071