HARRIS SEMICONDUCTOR
A DIVISION OF HARRIS CORPORATION

APPLICATION NOTE 204

DESIGNING WITH THE HD-0165 KEYBOARD ENCODER

BY D. F. JONES

The Harris Semiconductor type HD-0165 Keyboard Encoder integrated circuit provides an ideal low cost means of transforming signals from any manual keyboard into a TTL compatible parallel binary code.

CODE IMPLEMENTATION

The organization and truth table for the HD-0165 are shown in Figure 1 and Table 1. Basically, there are sixteen input lines, only one of which is normally connected at a time through the keyboard to the +5V supply. There are four output lines which produce a parallel binary code which is determined by the actuated input lines. The outputs will drive any DTL or TTL circuits with a fanout of six normal loads. The function of the two auxiliary outputs will be explained later.

To produce a certain output code by actuating a particular key, simply look down the output columns (1–4) of the truth table until the desired four-bit code is found; then look across to see which input is high (H) and connect that input pin to the key.

One fact which should be obvious is that since all sixteen possible combinations of four bits are available on the output lines, the numbering system for the input and output lines is arbitrary. The ordering of the truth table arbitrarily shows an ascending negative logic (H=0, L=1) output in binary code with output line 4 the most significant bit. But we do not need to be governed by this. If we want the "zero" key to produce LLLL, we do not need to tie input 1 to that key and put inverter gates in series with the outputs; we can simply tie input 16 to the "zero" key. Similarly, we could wire up a ten button keyboard to yield a 1–2–4–8 BCD code, a 1–2–4–2 BCD code, a 1–2–2–5 code, a Gray code, or any other code up to four bits simply by using the truth table and wiring the inputs to the appropriate keys. Redundant codes can be generated, if desired, by wiring one input line to more than one key.

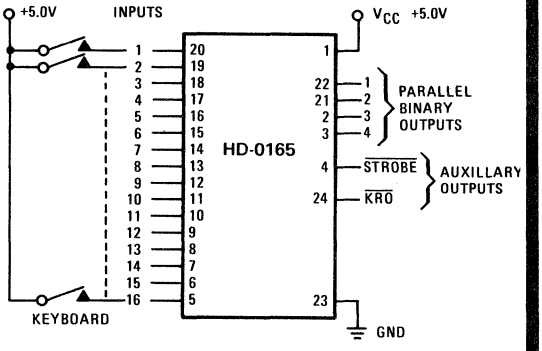

Figure 1. Keyboard Encoder Organization

ENCODING MORE THAN SIXTEEN KEYS

Two keyboard encoder circuits may be used to encode up to 256 keys, represented by eight bits.

The general scheme is to let the outputs from one device represent the four most significant bits and the output from the other device represent the four least significant bits of the output word. Each key is wired to one input on each of the two devices to produce the desired output code.

It is necessary to isolate the two wires from each key, since each device input usually is connected to more than one key. This is accomplished either by using double-pole key switches, or by using two diodes per key in series with the device input lines. The general scheme for cascading two encoders is shown in Figure 2.

TABLE I
TRUTH TABLE, HD-0165

INPUTS																OUTPUTS					
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	1	2	3	4	St.	K_{RO}
L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	H	H
H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	L	H
L	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H	H	L	H
L	L	H	L	L	L	L	L	L	L	L	L	L	L	L	L	H	L	H	H	L	H
L	L	L	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H	L	H
L	L	L	L	H	L	L	L	L	L	L	L	L	L	L	L	H	H	L	H	L	H
L	L	L	L	L	H	L	L	L	L	L	L	L	L	L	L	L	H	L	H	L	H
L	L	L	L	L	L	H	L	L	L	L	L	L	L	L	L	H	L	L	H	L	H
L	L	L	L	L	L	L	H	L	L	L	L	L	L	L	L	L	L	L	H	L	H
L	L	L	L	L	L	L	L	H	L	L	L	L	L	L	L	H	H	H	L	L	H
L	L	L	L	L	L	L	L	L	H	L	L	L	L	L	L	L	H	H	L	L	H
L	L	L	L	L	L	L	L	L	L	H	L	L	L	L	L	H	L	H	L	L	H
L	L	L	L	L	L	L	L	L	L	L	H	L	L	L	L	L	L	H	L	L	H
L	L	L	L	L	L	L	L	L	L	L	L	H	L	L	L	H	H	L	L	L	H
L	L	L	L	L	L	L	L	L	L	L	L	L	H	L	L	L	H	L	L	L	H
L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	L	H	L	L	L	L	H
L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	L	L	L	L	L	H
ANY TWO OR MORE HIGH																X	X	X	X	L	L

INPUTS: L = Open Circuit or < +1.0V H = > +4.5V Current Source
OUTPUTS: L = >+0.4V H = > +2.4V X = Erroneous Data

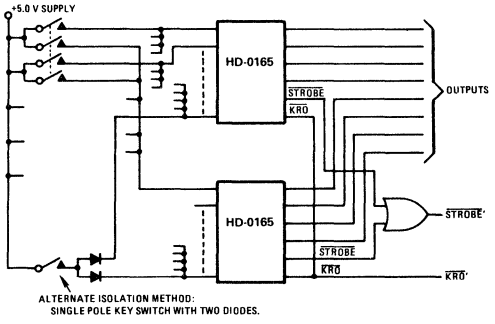

Figure 2

An important fact – when diode input isolation is used the diode forward voltage drop decreases the input current available to the encoder, and the result is that the encoder output fanout should be reduced to two TTL loads. The fanout capability could be restored by returning the keys to a higher voltage (+5.7V to +6.0V). This voltage could be generated by placing a high-current diode between the normal supply regulator output and its sense line, as shown in Figure 3.

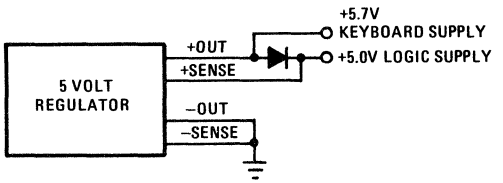

Figure 3

AUXILIARY OUTPUTS

As shown in the truth table, the STROBE output is high if no keys are actuated, and becomes low when any key is actuated. This signal is normally used to gate the encoded signal into other circuitry, and may be conditioned to eliminate any effects of switch bounce or skew in output propagation delays.

The KEY ROLLOVER (K_{RO}) output is normally high, and goes low only when two or more keys are actuated simultaneously. This signals that the output data word is not valid during that time.

Figure 4 illustrates one method of switch bounce surpression which also gives correct data input under key rollover conditions. One dual retriggerable monostable multivibrator device is required per keyboard to suppress multiple strobe pulses resulting from key bounce or arcing on either "make" or "break". With this scheme, there is no necessity for the use of mercury-wetted or other exotic key switches to prevent bounce effects.

The strobe and K_{RO} outputs from one or more encoders are combined at point A; so that the signal at point A is high when any one key is depressed, and low when no key or two or more keys are depressed.

The time delay, T_1, generated by the first monostable must be longer than the worst expected interval between switching spikes –

about 3 to 10 milliseconds is usually sufficient.

The time delay, T_2, generated by the second monostable depends on system requirements – about 0.5 microseconds will result from the values shown.

The waveforms shown at A, B, and C show a typical situation. At interval I, a key is depressed and the first monostable generates a delay, T_1, sufficiently long to allow any switch bounce spikes to die out. At the trailing edge of T_1, the second monostable generates a strobe pulse, T_2, for entering the data into the system. The delayed strobe also assures that the parallel data are stabilized when strobed into the system.

At interval II, a second key is depressed while the first key is not released until interval III. The first monostable might be triggered at this time, but this will not cause an output at C, because the clear (C_D) terminal of the second monostable will be held low. At interval III, when the first key is released, the two monostables will again be triggered, entering the data from the second key into the system.

If noise occurs at interval IV, when the second key is released, no signal will appear at C since C_D will again be low before the trailing edge at B is generated.

For data entry, it is necessary that a key be held down for longer than T_1, but since this interval is only a few milliseconds, it is highly unlikely that any deliberate key depression would be for a shorter interval.

The delayed strobe pulse, S_t, is used to signal the presence of new data to the system. The pulse can be used to gate a latch circuit or, if a serial data format is required, the pulse can be used to enable the parallel loading of a shift register.

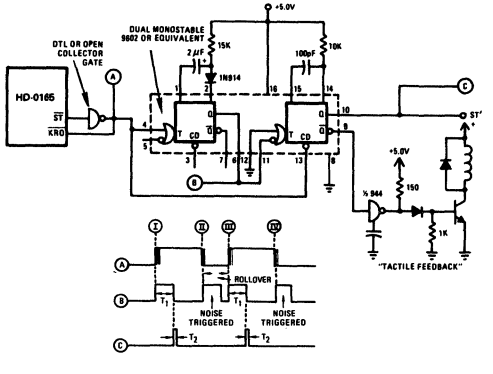

Figure 4. Switch Bounce Elimination

An optional "tactile feedback" circuit is also shown in Figure 4, which for a small increase in cost will greatly increase operator accuracy at high entry speeds. The transistor load is either a small solenoid hammer which taps the keyboard area, or a small loudspeaker which produces an audible "tick" each time data is entered. Since the "tick" only occurs when data entry actually occurs in the system, this scheme will give more accuracy (probably at lower cost) than special "tactile feel" key switches. Actual circuitry will depend on the type of load used. A capacitor is shown at the gate expander terminal to stretch the pulse at this point.

BATTERY OPERATED KEYBOARD

The circuit shown in Figure 5 will help conserve battery life in a portable keyboard. When no key is depressed, the PNP transistor is turned off, and there is no battery drain. When any key is pressed, the input current causes the transistor to turn on, supplying power to the encoder and to any other digital circuitry. The transistor should be capable of switching the necessary current and should have low $V_{CE(Sat)}$ for low power dissipation.

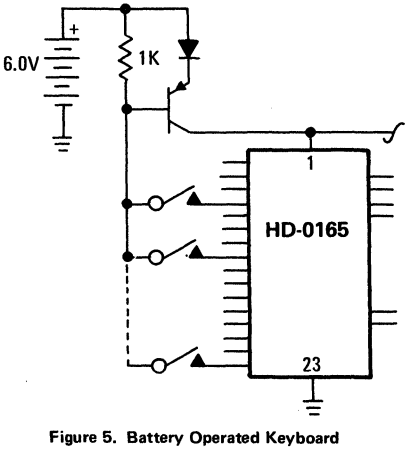

Figure 5. Battery Operated Keyboard

DESIGN EXAMPLE: TELETYPEWRITER KEYBOARD ENCODER

Each keyboard encoding design task must start with an analysis of the code to be produced and the arrangement of the keyboard. A minor complication arises when a single key must produce more than one code; for example when SHIFT and CONTROL keys are used.

7

As an example we will design a keyboard encoder for a teletypewriter or CRT terminal using the standard ASCII code for capital letters only, plus symbols and control functions.

First, for convenience, using the encoder truth table, we will assign a hexidecimal character to each device input line. The conventions for the output will be: positive logic (L=0, H=1); output line 1 will be the L.S.B., and output line 4 will be the M.S.B. This results in Table II.

Now, we tabulate the desired output words as a two digit hex code (Table III) with the first digit representing the four most significant bits and the second digit representing the four least significant bits. This now gives us the wiring list to correct each key in the normal (unshifted) mode to the two encoder circuits as in Figure 2. Since the hex code for "A" is CI, we can connect the two wires from the "A" key to the C input (Pin 17) of the top encoder and to the 1 input (Pin 6) of the bottom encoder of Figure 2. The rest of the keys are wired similarly for the specified output code in the unshifted state.

TABLE II
HEX CHARACTER ASSIGNMENT

HEX CHARACTER	INPUT LINE	PIN NUMBER
0	16	5
1	15	6
2	14	7
3	13	8
4	12	9
5	11	10
6	10	11
7	9	12
8	8	13
9	7	14
A	6	15
B	5	16
C	4	17
D	3	18
E	2	19
F	1	20

TABLE III
ASCII TELETYPEWRITER CODE

CHARACTER	HEX	CHARACTER	HEX	CHARACTER	HEX	CHARACTER	HEX
@	C0	Blank	A0	NULL	80	ACK	FC
A	C1	!	A1	SOH	81	ALT	FD
B	C2	"	A2	STX	82	ESC	FE
C	C3	#	A3	ETX	83	Rubout	FF
D	C4	$	A4	EOT	84		
E	C5	%	A5	WRU	85		
F	C6	&	A6	RU	86		
G	C7	'	A7	BELL	87		
H	C8	(	A8	FE	88		
I	C9	)	A9	HT	89		
J	CA	*	AA	LF	8A		
K	CB	+	AB	VT	8B		
L	CC	,	AC	FF	8C		
M	CD	–	AD	CR	8D		
N	CE	.	AE	SO	8E		
O	CF	/	AF	AI	8F		
P	D0	0	B0	DC0	90		
Q	D1	1	B1	X-ON	91		
R	D2	2	B2	Tape-ON	92		
S	D3	3	B3	X-OFF	93		
T	D4	4	B4	Tape-OFF	94		
U	D5	5	B5	ERR	95		
V	D6	6	B6	SYN	96		
W	D7	7	B7	LEM	97		
X	D8	8	B8	S_0	98		
Y	D9	9	B9	S_1	99		
Z	DA	:	BA	S_2	9A		
[	DB	;	BB	S_3	9B		
\	DC	<	BC	S_4	9C		
]	DD	=	BD	S_5	9D		
↑	DE	>	BE	S_6	9E		
←	DF	?	BF	S_7	9F		

TABLE IV

KEY			ENCODER INPUT	
NORMAL	SHIFT	CONTROL	MSB	LSB
A		SOH	C	1
B		STX	C	2
C		ETX	C	3
D		EOT	C	4
E		WRU	C	5
F		RU	C	6
G		BELL	C	7
H		FE	C	8
I		HT	C	9
J		LF	C	A
K	[	VT	C	B
L	\	FF	C	C
M	]	CR	C	D
N	↑	SO	C	E
O	←	SI	C	F
P	@	NUL	D	0
Q		X-ON	D	1
R		Tape-ON	D	2
S		X-OFF	D	3
T		Tape-OFF	D	4
U		ERR	D	5
V		SYN	D	6
W		LEM	D	7
X		S_0	D	8
Y		S_1	D	9
Z		S_2	D	A

KEY			ENCODER INPUT	
NORMAL	SHIFT	CONTROL	MSB	LSB
1	!		B	1
2	"		B	2
3	#		B	3
4	$		B	4
5	%		B	5
6	&		B	6
7	'		B	7
8	(		B	8
9	)		B	9
0			B	0
:	*		B	A
–	=		A	D
.	>		A	E
,	<		A	C
;	+		B	B
/	?		A	F
Line Feed			8	A
Return			8	D
Rubout			F	F
Space			A	0
ACK			F	C
ALT			F	D
ESC			F	E

For this example, we will design one of the more popular terminal keyboard arrangements as shown in Table IV.

Comparing Table IV with Table III, we note that the L.S.B.'s change in a particular pattern: A→B, B→A, C→D, D→C. Further investigation shows that this is simply an inversion of the fifth output bit. So, we could make the SHIFT key control a group of gates in series with the fifth output bit to pass this bit either inverted or not inverted.

Investigation of the control mode codes shows that bits 6 and 7 must always be "0" when the CONTROL key is depressed, the other bits remaining the same as in the normal mode for that key. This can easily be implemented by connecting DTL gates in "wired-OR" to these outputs. Control functions S_3 through S_7 can be generated by holding both the SHIFT and CONTROL keys down and pressing "L" through "P", respectively; or these could be implemented by additional wiring.

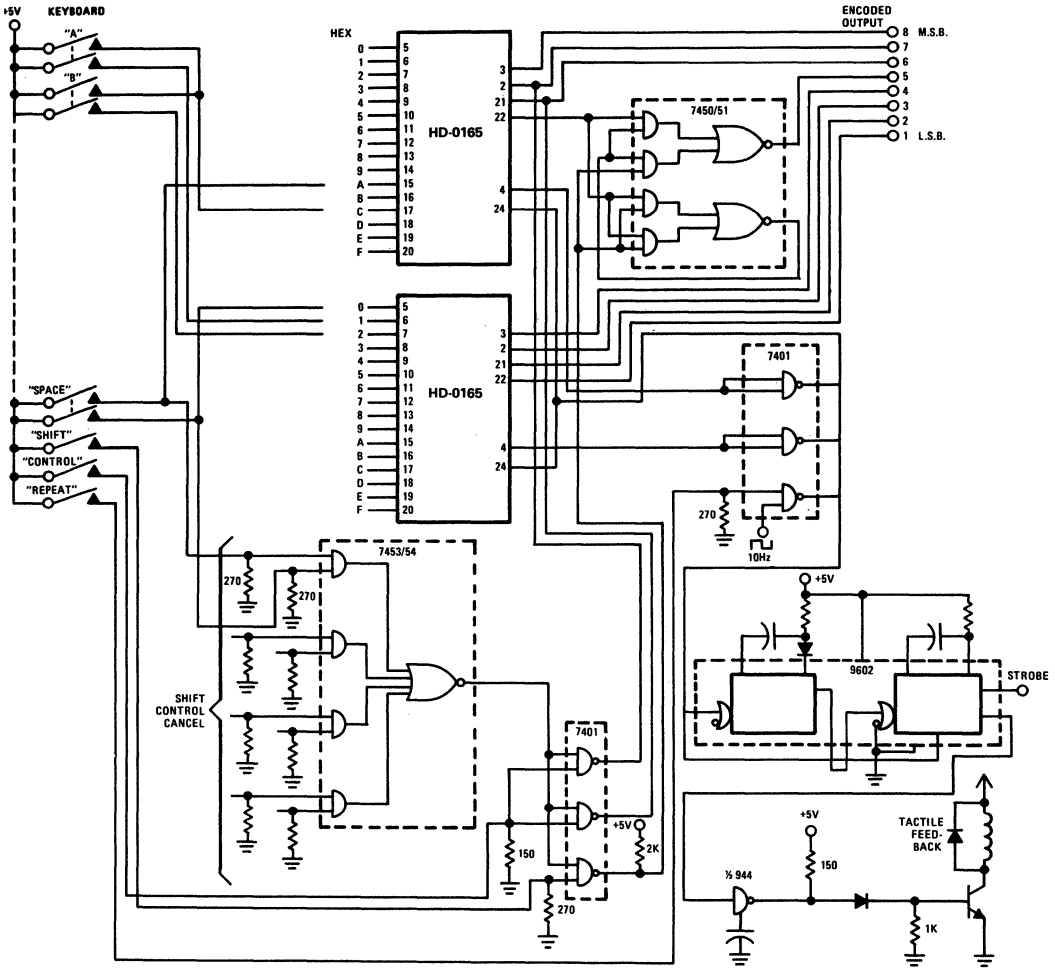

Figure 6. Teletypewriter (ASCII) Encoder

Figure 6 shows the finished design, with the wiring from keys to encoder input indicated in Table IV.

It may be desirable to have some keys, such as the SPACE bar, produce the same output in all three modes. If the "cancel" inputs shown in Figure 6, are wired to those keys, the effect of the SHIFT and CONTROL keys will be nullified for those particular keys.

An optional feature shown is the REPEAT key which pulses the strobe output at about 10cps as long as it and another key are held down.

A full typewriter ASCII keyboard with upper and lower case letters can be implemented in a similar fashion. The "shift" requires inversion of either bit 5 or bit 6 depending on the state of bit 7, so the logic at the output is somewhat more complex.

UNIVERSAL KEYBOARD ENCODER

Suppose we required a keyboard with a code in which there was no simple logical relationships between shifted and unshifted output words—or a keyboard which could produce several entirely different codes at the flip of a switch—or one which could be supplied with any desired code on very short notice. Any of these problems can be readily solved by using keyboard encoder circuits to generate a universal code—which can then be translated to the desired output code using programmable read-only memories.

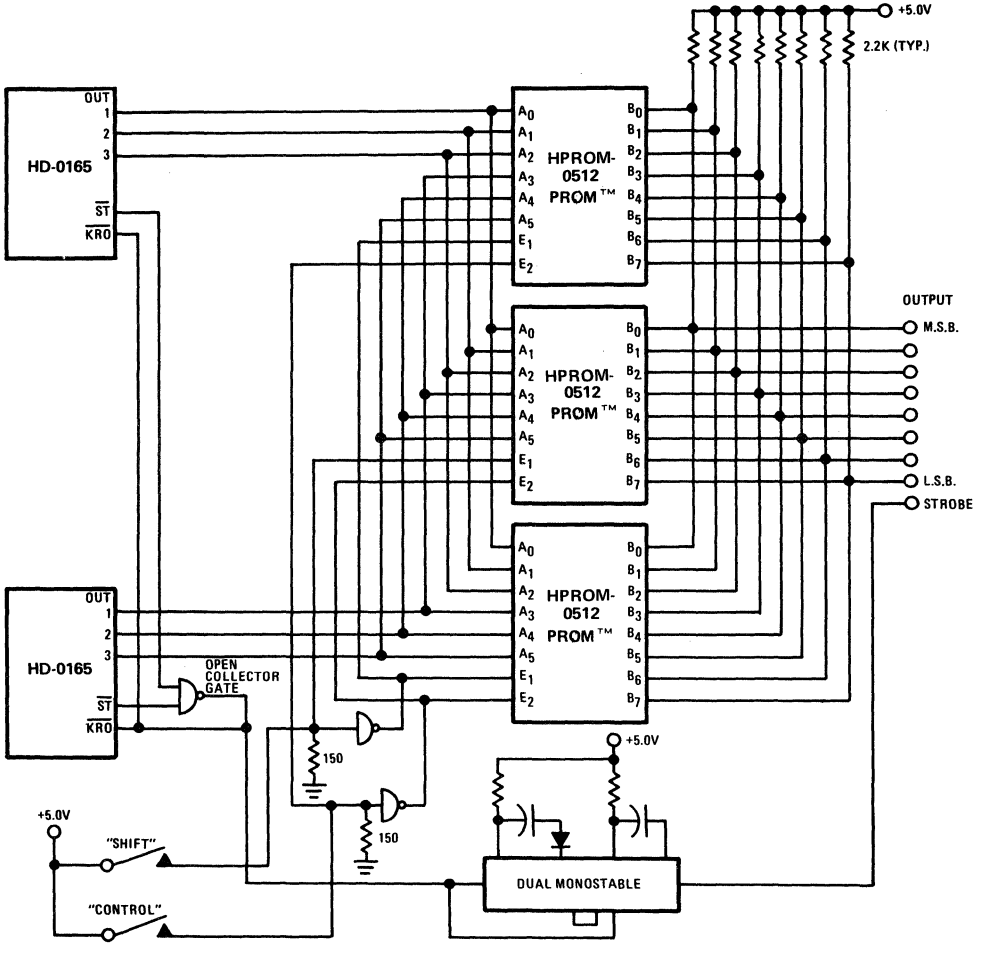

Figure 7. Universal Keyboard Encoder

Figure 7 shows the organization for a universal keyboard encoder. Up to 64 keys are wired to two encoder circuits which produce a six-bit output code (the fourth output bit of each encoder is not used). The key to encoder wiring is arbitrary as long as each key produces a unique six-bit output code from the encoders.

The six-bit code forms the address for one of three ROM's wired in parallel. Each HROM-0512 stores 64 eight-bit words, which can be programmed electrically by the user (contact Harris Semiconductor for data sheets and programming instructions).

Each ROM is initially programmed for the desired output word at the address location corresponding to a certain key. One ROM contains all 64 output words for the unshifted mode, the second contains all words for the shifted mode, and the third contains all words for the control mode. Selection of the particular ROM is accomplished through its enable inputs.

Obviously, there is no requirement for a logical relationship between corresponding words in the three ROM's, which greatly simplifies implementation of special codes. If it is desired that a particular word remain the same in all three modes, simply program the same word at the same address in the three ROM's.

A great advantage of this system is that any eight-bit code can be produced without any changes in wiring or P.C. board layout, simply by plugging in ROM's programmed to desired code. Additional ROM's could be wired in parallel with the enable inputs wired appropriately to produce multiple codes (both ASCII and EBCDIC codes, for example) either simultaneously or selectively.