Truth Tables

a) Address Latch

ADDRESS LATCH CONTROL	FUNCTION PERFORMED
0	Latched
1	Fall Through

b) Output

OUTPUT $\overline{\text{ENABLE}}$	STATE OF THE OUTPUT
1	Output Hi-Z
0	Data Out

c) 4-Bit Line Counter

CLOCK CONTROL	LINE CLOCK	$\overline{\text{CLEAR}}$	LINE COUNTER
H	⌐	H	Increment line counter
X	X	L	Asynchronous clear resets counter
L	X	H	Clock inhibited
H	¬	H	No change on high-to-low clock edge

X = Don't care

Definitions

A1–A6: Character address. A 6-bit code which selects 1 of the 64 characters in the font.

$\overline{\text{Clear}}$: Active low clear for mod 16 row counter, (can be used to truncate mod 16 counter).

Line Clock: Clock that advances the line counter. Advances counter on the low-to-high transition.

Clock Control: Enables line clock when high and disables line clock when low.

Load $\overline{\text{Enable}}$: Active low load command which routes data from the character ROM to the "D" inputs of the 7-bit shift register.

Dot Clock: A low-to-high transition of the dot clock loads the shift register if load $\overline{\text{enable}}$ is low or shifts data if load $\overline{\text{enable}}$ is high.

Output $\overline{\text{Enable}}$: An active low output enable. When high the output is in the Hi-Z state.

Output: A TTL TRI-STATE output buffer.

Functional Description

To select a character, a 6-bit binary word must be present at the address inputs A1–A6 when the address latch control is high. This address can be latched by bringing the address latch control signal low after a 40 ns set-up time. When the $\overline{\text{clear}}$ input receives a low pulse, the counter is reset to zero. The shift register can be loaded (T_{S2} ns) after the character is addressed. Data, representing one horizontal line of the addressed character, is available at the output when the load $\overline{\text{enable}}$ input is brought low. As shown in *Figure 1*, valid data arrives serially at the output. Dot clock pulses beyond that required to shift out one line of the character will add lows to the end of character. This pulses at the line clock input. Any additional line clocks beyond that required to display the character will put a vertical space between characters. This spacing can be truncated by bringing the $\overline{\text{clear}}$ input low. Detailed system application infomation is contained in application note AN-167 available from National.

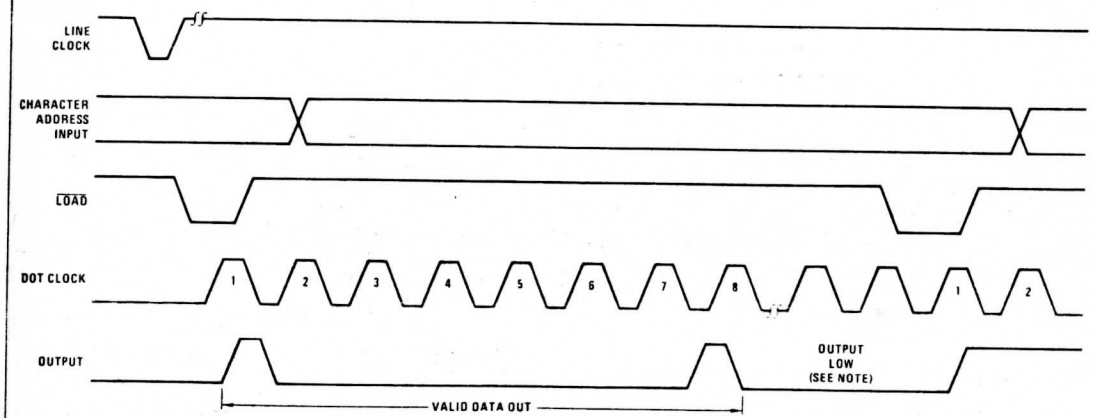
A two character display example is shown in *Figure 3* and a typical system timing waveform is shown in *Figure 4*.

A chip select input is provided for expansion of the character font. The various standard fonts are shown in

Functional Description (Continued)

Character Cycle – ROM data corresponding to one line of characters is loaded into the shift register T_{S2} after the ROM is addressed. When load $\overline{\text{enable}}$ goes low, ROM data is allowed to be present at the D input of the shift register via the MUX. The first bit of the ROM data is transferred to the output at the next low-to-high transition of the dot clock. After load $\overline{\text{enable}}$ goes back high, the second to seventh clock pulses shift out the rest of the selected row of the addressed character. Additional clock pulses will shift out low data used for spacing.

Line Cycle – The line counter is a mod 16 counter. A low-to-high transition of the line clock advances the line counter to the next count. If, for any reason, the counts need to be truncated, a low signal at the $\overline{\text{clear}}$ input resets the counter to zero. The clock control may be used as a line clock disable. A high signal at the line clock control terminal enables the counter and a low signal disables the line clock.



Note. Output goes and stays low following the leading edge of the eighth Dot-Clock pulse until Load $\overline{\text{enable}}$ is enabled again and new parallel data is loaded into the shift register.

FIGURE 1. Character Cycle

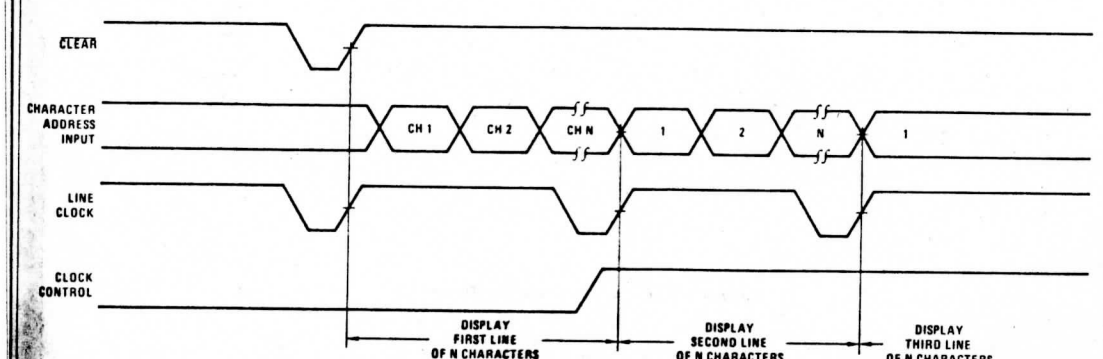


FIGURE 2. Line Cycle