



# Complex Standards

## MM5725 one chip calculator

### general description

The MM5725 is an 8-digit calculator on one MOS/LSI chip. It uses three registers to provide the four arithmetic functions (+, -, x, and  $\div$ ). The calculator has a 16-place decimal point register and self-contained oscillator and clock driver. In addition to operating the machine the internal clock provides timed key-bounce protection. Calculations are handled in floating point in both entry and result display. Results are right justified to eight significant digits in all calculations. Leading zeros are suppressed.

The 8-digit display is automatically multiplexed and presented in standard 7-segment form. Eight digit select output signals ( $T_1$  through  $T_8$ ) appear sequentially from least to most significant, while the segment outputs ( $S_a$  through  $S_g$ , and DP) provide the decoded figures and the decimal point. Minus sign indication appears on a separate line, at the same time as the least significant digit.

The eight digit select lines are also used to scan the keyboard, which is arranged as an 8 x 4 array (see Figure 2); four lines return the keyboard entries into the chip.

### features

- 8 Digit multiplexed display
- Four functions (+, -, x, and  $\div$ )
- Full floating point input
- Full floating point output
- True credit balance
- Chain calculations
- Overflow indication
- Leading zero suppression
- Trailing zero suppression in division
- Decimal point memory to 16 digits
- Seven segment decoder on chip
- On chip clock oscillator and driver
- Key-bounce elimination

### key description

**Clear:** should be depressed twice when power is first turned on. A single zero and decimal point will then be displayed in the extreme right digit position. In all further operations, a digit will always be displayed in at least that position. During operations, the clear key combines the function of a "clear entry" and "clear accumulator." A first depression clears the display of all digits entered since the last calculation, and will restore the result of the last calculation, if any, to the display. If no prior result is present in the accumulator, a single zero without decimal point will be displayed.

A second depression will clear the accumulator and display, leaving a single zero and decimal point in the extreme right digit position.

**Numerics:** are entered from the right. If a previous result exists in the accumulator, it will remain in there while the first numeric entry clears it from the display, and appears in its place. At any point in the entry of the string of numerics, the user may insert a decimal point. Once a *decimal point* is entered, it remains displayed to the right of the units digit. Additional decimal point entries and any numerics after the eighth are disregarded. No overflow can result on input.

When the new number has been fully entered, an operation may be selected. The operator key (+, -, x,  $\div$ ) is selected by the user after the operands (accumulator and display) are present in the machine. The operations will now be described in detail.

**Add/Enter:** adds the display register to the accumulator, enters the sum into the accumulator, and displays the new accumulator content. If the accumulator was empty, it enters the displayed number into the accumulator, and continues to display the same number.

**Subtract:** subtracts the display register from the accumulator, stores the result in the accumulator, and displays the new accumulator content. If the new result is negative, the minus sign indicator lights, and will remain lit so long as the accumulator remains negative.

**Multiply:** multiplies the accumulator by the display register, stores the result in the accumulator and displays it. If the product exceeds 99,999,999, its eight most significant digits will be displayed without a decimal point. Further multiplication of the accumulator is allowable if the result will not exceed  $10^{16} - 1$ . If a product exceeds this value, the machine will *overflow*, lose all content, and display all zeroes with all decimal points.

**Divide:** divides the accumulator by the display register, stores the result in the accumulator, and displays it. If the result of a multiplication, or of a division by a number less than one, results in a display with no decimal point, the point may be located by successive division by powers of ten. (It is often convenient to divide by 1,000 to shift a result three decimal points at a time.) Unlike some less sophisticated designs, the MM5725 keeps results of divisions as far to the right in the display as possible (see division examples *b* and *c* below) and so gives indication of a result with a remainder more than eight orders of magnitude below the most significant digit in a result. The remainder is not directly retrievable.

Order Number MM5725N  
See Package 19

### absolute maximum ratings

Voltage at any pin except $V_{GG}$ relative to $V_{SS}$ (all others GND)	+0.3V to -28V
Voltage at $V_{GG}$ pin relative to $V_{SS}$ (all others GND)	+0.3V to -38V
Voltage at pins $T_1$ through $T_8$ relative to $V_{SS}$ (with $V_{DD} = -28V$ )	+0.3V to -55V
Operating Temperature Range	0°C to 70°C
Storage Temperature Range	-55°C to 150°C

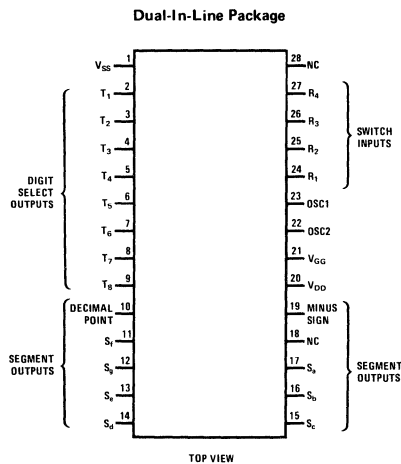
### operating voltage range

$V_{SS} = 0V$   
 $V_{DD} = -28V \pm 7\%$   
 $V_{GG} = -35V \pm 7\%$

### electrical characteristics

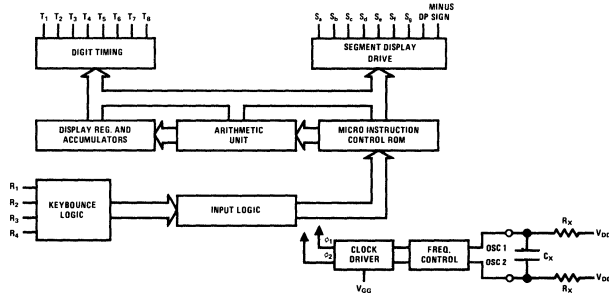
PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Current ( $I_{DD}$ )	$\left\{ \begin{array}{l} V_{DD} = -28V \\ V_{GG} = -35V \end{array} \right\}$		10.0	18.0	mA
Supply Current ( $I_{GG}$ )				2.0	2.7
Switch Input Current ( $R_1 - R_4$ ), (Internal Resistor to $V_{DD}$ )	$0 \leq V_{IN} \leq -30V$			100	$\mu A$
Digit Select Output Current ( $T_1 - T_8$ ), (open drain)	$V_{OUT} = -8.0V, V_{DD} = -28V$	4	5		mA
Segment Output Current ( $S_a - S_g$ , MS, DP), (push-pull)	$V_{OUT} = -2.0V$ (Segment on)	-300	-500		$\mu A$
	$V_{OUT} = -10.0V$ (Segment off)	+300	+500		$\mu A$

### connection diagram

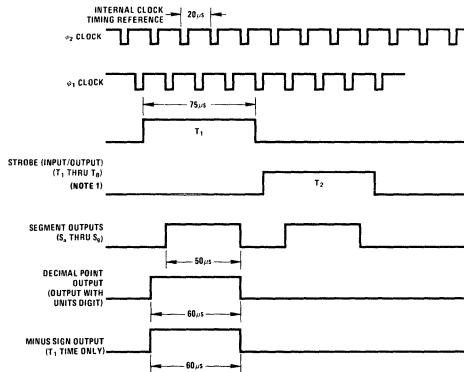


10

# calculator chip system organization



## timing diagram



Note 1: Used to strobe keyboard for input conditions, and to multiplex digit outputs for display.  
 Note 2: Times shown are nominal.

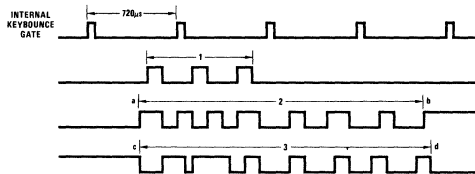
## keybounce protection

Keybounce protection in the MM5725 is ensured by a series of latches and interlocks operated from the internal clocking scheme. The protection

intervals are derived from the scan period, nominally  $720\mu s$ , and are shown in the table.

TABLE. Worst Acceptable Switch Characteristics

CHARACTERISTIC	SCAN PERIODS	TYPICAL TIME
Widest discrete noise burst which will not cause false data entry	1	.72 ms (1)
Longest switch bounce tolerable as switch closes	3	2.16 ms (2)
Longest switch bounce tolerable as switch opens	3	2.16 ms (3)



- a. Point where switch first closes
- b. Point where switch must be fully closed
- c. Point where switch first opens
- d. Point where switch must be fully open



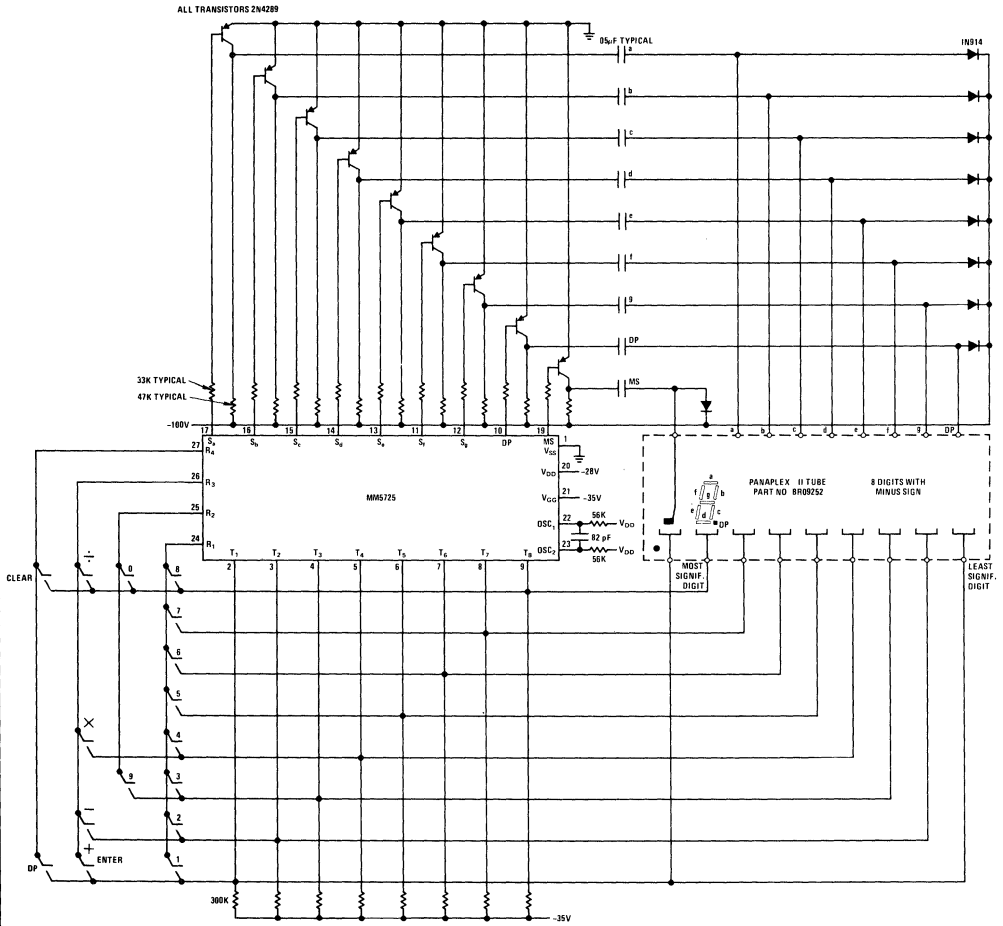


FIGURE 2. MM5725 Interface to Panaplex II\* Neon Display Tube

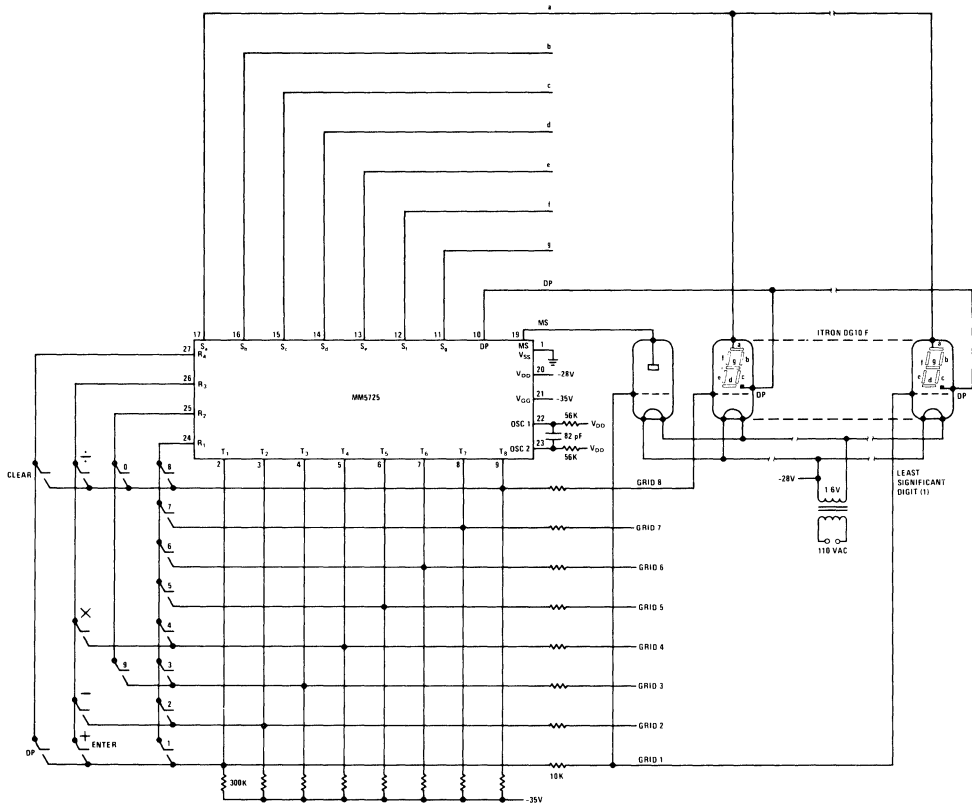


FIGURE 3. MM5725 Interface to Fluorescent Vacuum Display Tubes

performance curves

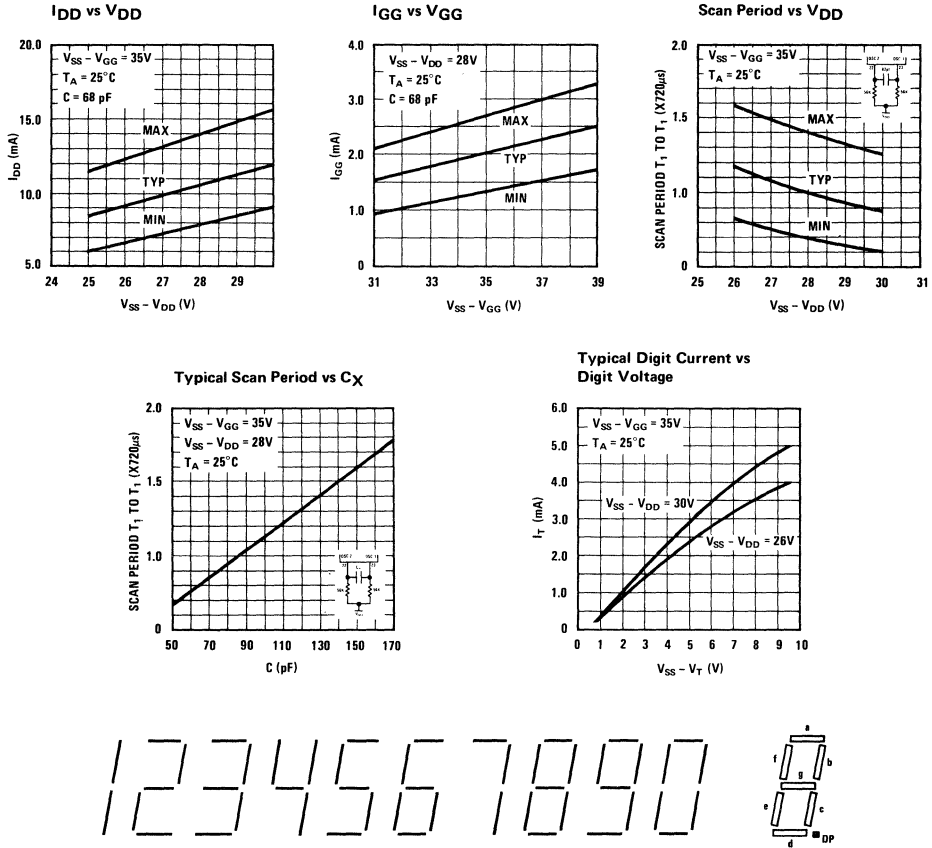


FIGURE 4. MM5725 Display Font

accounting machine notation

The MM5725 performs its arithmetic according to a logic form known as Polish notation. In this notation, as pointed out in the description on page one, the new number keyed into the machine operates on the previously accumulated result only when the operator key (+, -, x,  $\div$ ) is struck. And at that point, the result appears. There is no separate equals (=) key, as all four operator keys perform the function of generating the result of a calculation. The use of an operator key after the data entry is a very natural mode of operation in, for instance, credit balance transactions. For calculator users without previous training, as one might typically classify first-time low-cost calculator buyers, it is equally natural to extend the Polish notation to the multiply and divide operations.

Users who have been exposed to electric accounting machines, however, may be confused by this routine. Those machines usually use a hybrid notation, Polish in addition and subtraction, but

algebraic in multiplication and division: for example  $A \times B =$ . The MM5725 may be used in such machines by adding a key, wired in parallel to the +/Enter key, called ( $\frac{x}{\div}$ ) and by marking the existing multiplication and division keys  $\frac{x}{\div}$  and  $\frac{\div}{\times}$ .

The operation of  $A \times B = C$  may now be compared

Polish Entry	"Pseudo-Hybrid" Key Caps
2	2
+/Enter	$\frac{x}{\div}$ Actually causes entry
3	3
x 6 appears	$\frac{x}{\div}$ 6 appears

and  $E \div F = G_T$  looks like this:

8	8
+/Enter	$\frac{x}{\div}$
4	4
$\div$ 2 appears	$\frac{\div}{\times}$ 2 appears

**sample calculations**

CALCULATION	KEY PROCEDURE	DISPLAY									
		-	8	7	6	5	4	3	2	1	
<b>Entry</b> Initial setting after clear operation Enter number 37.48  Result	3 7 . 4 8									0. 3. 7. 3 7. 3 7. 4 3 7. 4 3 7. 4 8	
<b>Addition</b> a) $3.1 + 4.11 = 7.21$  Result	3 . 1 + (Enter) 4 . 1 1 +									0. 3. 3. 3. 1 3. 1 4. 4. 4. 1 4. 1 1 7. 2 1	
b) $0 + 4.11 = 4.11$  Result	4 . 1 1 +									0. 4. 4. 4. 1 4. 1 1 4. 1 1	
c) $-7.032 + 3 = -4.032$  Result	7 . 0 3 2 - 3 +									0. 7. 7. 7. 0 7. 0 3 7. 0 3 2 7. 0 3 2 3. 4. 0 3 2	
d) $4.0053114 + 8 = 12.0053114$ Assume 4.0053114 result from previous calculation  Result	. 8 +		4.	0	0	5	3	1	1	4	8. 1 2. 0 0 5 3 1 1
<b>Subtraction</b> a) $4 - 2.1 = 1.9$  Result	4 + (Enter) 2 . 1 -									0. 4. 4. 2. 2. 2. 1 1. 9	



sample calculations (con't)		
CALCULATION	KEY PROCEDURE	DISPLAY
		- 8 7 6 5 4 3 2 1
b) $0 - 3.97 = -3.97$	3 . 9 7 - Result	- 8 7 6 5 4 3 2 1 0. 3. 3. 9 3. 9 7 3. 9 7
.c) $-4.6 - 3.11 = -7.71$	4 . 6 - 3 . 1 1 - Result	- 8 7 6 5 4 3 2 1 0. 4. 4. 4. 6 4. 6 3. 3. 3. 1 3. 1 7. 7 1
<b>Multiplication</b> a) $4 \times 5 = 20$	4 + (Enter) 5 x Result	- 8 7 6 5 4 3 2 1 0. 4. 4. 5. 2 0.
b) above result $\times 5 = 100$	5 x Result	- 8 7 6 5 4 3 2 1 5. 1 0 0.
c) $-49.6 \times 0.42 = -20.832$	4 9 . 6 - . 4 2 x Result	- 8 7 6 5 4 3 2 1 0. 4. 4 9. 4 9. 4 9. 6 4 9. 6 0. 0. 4 0. 4 2 2 0. 8 3 2
d) $8765.432 \times 100. = 876543.20$ assume first digit result from previous calculation	1 0 0 x Result	- 8 7 6 5 4 3 2 1 8 7 6 5 4 3 2 0 1. 0. 1 0 0. 8 7 6 5 4 3 2 0
<b>Division</b> a) $4 \div 3 = 1.3333333$	4 + (Enter) 3 ÷ Result	- 8 7 6 5 4 3 2 1 0. 4. 4. 3. 1. 3 3 3 3 3 3 3



sample calculations (con't)									
CALCULATION	KEY PROCEDURE	DISPLAY							
		-	8	7	6	5	4	3	2
<b>Chain Calculation</b> a) $3.7 + 4.85 - 3.49 = 5.06$	3 . 7 + (Enter) 4 . 8 5 + 3 . 4 9 -								0. 3. 3. 7 7 4. 4. 8 5 5 3. 3. 4 9 6
b) $472 \times .018 + 3.1 = 11.596$	4 7 2 + (Enter) . 0 1 8 x = 3 . 1 +								0. 4. 7. 2. 2. 0. 0 1 8 6 3. 3. 1 6
c) $\frac{(246 - 37.8) \times 4.5}{9} = 104.1$	2 4 6 + (Enter) 3 7 . 8 - 4 . 5 x 9 ÷								0. 2. 4. 6. 6. 3. 7. 7. 8 2 4. 4. 5 9 9. 1
Result									

**sample calculations (con't)**

CALCULATION	KEY PROCEDURE	DISPLAY								
		-	8	7	6	5	4	3	2	1
d) $\frac{266475 \times 8624}{187.9} = 12230337.$					2	6	6	4	7	5.
Assume first number from previous result	8									8.
	6								8	6.
	2							8	6	2.
	4						8	6	2	4.
Note that decimal point capacity of display has been exceeded, but is stored in memory.	x	2	2	9	8	0	8	0	4	
	1									1.
	8								1	8.
	7							1	8	7.
	.							1	8	7.
	9					1	8	7.	9	
Result	÷	1	2	2	3	0	3	3	7.	
Decimal point appears										