Here is a simple program using the 8 pin PIC12C508 microcontroller that does the job of a programmable 4-bit Mode N up counter. The chip can be used to replace a TTL or CMOS 4-bit counter. Also, the counter can be configured as mode N at the time of power on. The counter can be loaded with a binary value given to D0-D3 during operation, by keeping the Preset pin High.

The design utilizes all the pins available. No additional components are required for implementing the design. Currently, the available presettable counter comes with at least 14 pins to achieve this and also to use a 4-bit counter as BCD or Modulo N counter needs external gates. The circuit can work with a maximum clock rate of 50 kHZ.

The chip operating as the counter has the pin configuration as shown below.

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Pin Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VDD</td>
</tr>
<tr>
<td>2</td>
<td>Clock Input (Active Low)</td>
</tr>
<tr>
<td>3</td>
<td>D3</td>
</tr>
<tr>
<td>4</td>
<td>Preset (Active High)</td>
</tr>
<tr>
<td>5</td>
<td>D2</td>
</tr>
<tr>
<td>6</td>
<td>D1</td>
</tr>
<tr>
<td>7</td>
<td>D0</td>
</tr>
<tr>
<td>8</td>
<td>VSS</td>
</tr>
</tbody>
</table>

**Note 1:** D0 to D3 are the data pins acting as output pins during counting and acts as inputs during Parallel loading and also on Mode set.

**Note 2:** Grounding the Clock input pin (pin2) and Preset pin (pin 4) at power on reads the pins d0 to d3 and mode is set as the binary value given to d0 to d3 (do is LSB).

The design is implemented as an up counter. It can be easily configured as a down counter by simply changing only one instruction in the software (increment instruction to decrement instruction.).

**APPLICATION OPERATION**

The operation of the counter is simple. No additional components are needed. It can replace a 4-bit programmable counter. The I/O pins are configured as follows:

<table>
<thead>
<tr>
<th>PIC12C5XX pin Name</th>
<th>Application pin Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP0</td>
<td>D0</td>
</tr>
<tr>
<td>GP1</td>
<td>D1</td>
</tr>
<tr>
<td>GP2</td>
<td>D2</td>
</tr>
<tr>
<td>GP4</td>
<td>D3</td>
</tr>
<tr>
<td>GP5</td>
<td>Clock In</td>
</tr>
<tr>
<td>GP3</td>
<td>Preset in</td>
</tr>
<tr>
<td>VDD</td>
<td>Vcc</td>
</tr>
<tr>
<td>VSS</td>
<td>GND</td>
</tr>
</tbody>
</table>

At power on the PIC12C508 initialized and checks if both Clock and preset inputs are in logic low. If both are held at logic low, then data available at D0 to D3 are read by the CPU, and the mode is set accordingly. After this, the CPU looks for the logic levels on clock and Preset and, accordingly, the program branches either to parallel load or to count mode.

One difficult problem that occurred in developing the program was that the Pin GP3 is only an input pin. Here, D3 is bi-directional and GP3 could not be used for it. Hence, it was decided to use GP4 as D3. This created problem in comparing data at inputs with data in registers for setting Modes as well as sending the count values to the output pins. A sub routine called BIT_CHANGE is included for this purpose of changing bit position (bit 4 to bit 3).

It requires a maximum 18 CPU clock cycles for one count. The counter will detect the next counter input clock only after 18 micro seconds if we use a 4MHZ CPU clock. Hence the maximum frequency of operation of the counter is limited to say 50 kHZ. However, this is sufficient for low speed application.

Similarly the output change occurs only a few CPU clock cycles after the counter clock input is activated. For this reason, counter implemented is not suitable as a synchronous counter. But it can be safely used for low frequency circuits as a synchronous counter with fixed propagation delay of several microseconds.
APPENDIX A: SOURCE CODE

; Program for Presettable Mode N counter.

COUNT EQU 08  
CHANGE EQU 09  
MODE EQU 0A  

;********test for initialization******
START BTFSS GPIO, 5 ; test for clock input (clock for 0)
BTFSS GPIO, 3 ; test for Preset input (pl for 0)
GOTO MODE_SET ;

;******counting program***********
COUNT_START BTFSC GPIO, 5 ; check if clock is active(low)
GOTO PL      ; check for parallel load if no clock
MOVF COUNT,0
MOVWF CHANGE
BCF CHANGE,4 ;
BTFSC COUNT,3 ;For interchanging bit3 with bit4
BSF CHANGE,4 ;
MOVF CHANGE,0
MOVF GPIO ; output the count to d0~d3
INCF COUNT, 1 ; count= COUNT+1
DECF MODE,0
XORWF COUNT,0 ;MODE CHECK
BTFSS STATUS,2 ;skip if SET
CLRF COUNT
GOTO COUNT_START

; ***************parallel load********************************************
PL BTFSS GPIO, 3 ; check for parallel Load
GOTO COUNT_START ; goto count mode if pl is not active
MOVFW ff ; config the d0 to d3 as inputs
TRIS GPIO ; Parallel load the value
CALL BIT_CHANGE
MOVF CHANGE
MOVWF COUNT
GOTO PL

;************************************************************************

MODE_SET MOVFW, ff ; make d0 to d3 as inputs
TRIS GPIO ; Read the mode N
CALL BIT_CHANGE
MOVF CHANGE
MOVWF MODE
GOTO START

BIT_CHANGE MOVFW GPIO
ANDLW 0F
MOVF CHANGE
BCF CHANGE,3
BFFSC CHANGE,4
BSF CHANGE,3
RETLW 00

;******END OF PROGRAM*******
EPROM AND RAM USAGE

Only 42 EPROM locations are needed for the program. The total RAM locations needed are only 10 bytes including general purpose registers.
NOTES: