PIC17C7XX

EPROM Memory Programming Specification

This document includes the programming specifications for the following devices:

- PIC17C752
- PIC17C756

1.0 PROGRAMMING THE PIC17C7XX

The PIC17C7XX is programmed using the TABLWT instruction. The table pointer points to the internal EPROM location start. Therefore, a user can program an EPROM location while executing code (even from internal EPROM). This programming specification applies to PIC17C7XX devices in all packages.

For the convenience of a programmer developer, a "program & verify" routine is provided in the on-chip test program memory space. The program resides in ROM and not EPROM, therefore, it is not erasable. The "program/verify" routine allows the user to load any address, program a location, verify a location or increment to the next location. It allows variable programming pulse width.

The PIC17C7XX group of the High End Family has added a feature that allows the serial programming of the device. This is very useful in applications where it is desirable to program the device after it has been manufactured into the users system (In-circuit Serial Programming (ISP)). This allows the product to be shipped with the most current version of the firmware, since the microcontroller can be programmed just before final test as opposed to before board manufacture. Devices may be serialized to make the product unique, "special" variants of the product may be offered, and code updates are possible. This allows for increased design flexibility.

1.1 Hardware Requirements

Since the PIC17C7XX under programming is actually executing code from "boot ROM," a clock must be provided to the part. Furthermore, the PIC17C7XX under programming may have any oscillator configuration (EC, XT, LF or RC). Therefore, the external clock driver must be able to overdrive pulldown in RC mode. CMOS drivers are required since the OSC1 input has a Schmitt trigger input with levels (typically) of 0.2 V_{DD} and 0.8 V_{DD}. See the PIC17C5X data sheet (DS30264) for exact specifications.

The PIC17C7XX requires two programmable power supplies, one for V_{DD} (2.5V to 6.0V recommended) and one for V_{PP} (13 ± 0.25V). Both supplies should have a minimum resolution of 0.25V.

The PIC17C7XX uses an intelligent algorithm. The algorithm calls for program verification at V_{DD}min as well as V_{DD}max. Verification at V_{DD}min guarantees good "erase margin". Verification at V_{DD}max guarantees good "program margin". Three times (3X) additional pulses will increase program margin then beyond V_{DD}max and insure safe operation in user system.

The actual programming must be done with V_{DD} in the V_{DDP} range (4.75 - 5.25V).

V_{DDP} = V_{DD} range required during programming.

V_{DD}min. = minimum operating V_{DD} spec. for the part.

V_{DD}max. = maximum operating V_{CC} spec for the part.

Programmers must verify the PIC17C7XX at its specified V_{DD}max and V_{DD}min levels. Since Microchip may introduce future versions of the PIC17C7XX with a broader V_{DD} range, it is best that these levels are user selectable (defaults are ok).

Note:	Any programmer not meeting these requirements may only be classified as "prototype" or "development" programmer but not a "production" quality programmer.

Pin Diagram

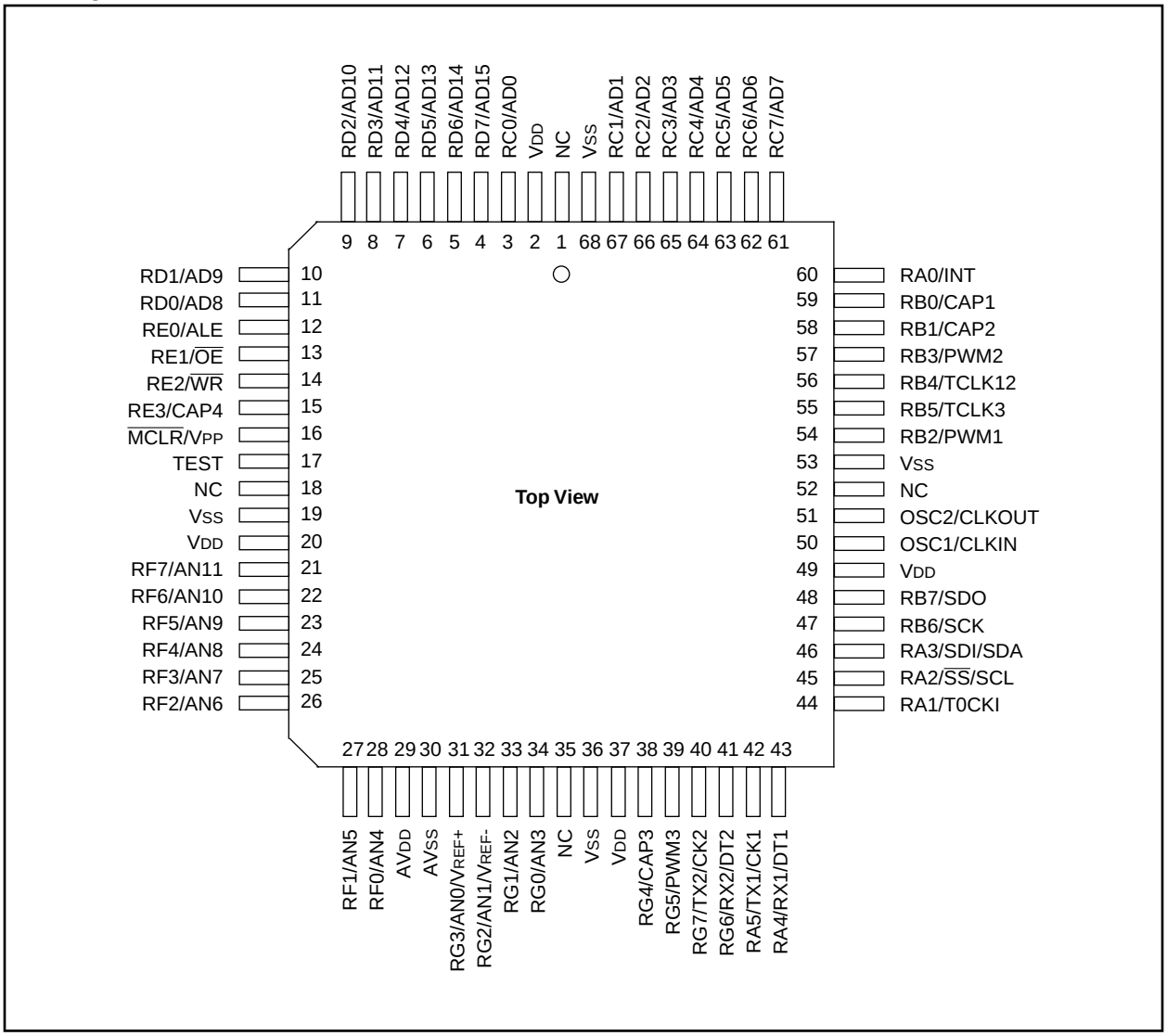


TABLE 1-1: PIN DESCRIPTIONS (DURING PROGRAMMING IN PARALLEL MODE): PIC17C7XX

Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
RA4:RA0	RA4:RA0	I	Necessary in programming mode
TEST	TEST	I	Must be set to “high” to enter programming mode
PORTB<7:0>	DAD15:DAD8	I/O	Address & data: high byte
PORTC<7:0>	DAD7:DAD0	I/O	Address & data: low byte
$\overline{MCLR}$/VPP	VPP	P	Programming Power
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground

Legend: I = Input, O = Output, P = Power

2.0 PARALLEL MODE PROGRAM ENTRY

To execute the programming routine, the user must hold TEST pin high, RA2, RA3 must be low and RA4 must be high (after power-up) while keeping $\overline{MCLR}$ low and then raise $\overline{MCLR}$ pin from V_{IL} to V_{DD} or V_{PP}. This will force FFE0h in the program counter and execution will begin at that location (the beginning of the boot code) following reset. Execution is forced to Internal mode by overriding the fuse configuration. The code protect bit is not overwritten. The program immediately polls PORTB<7:0> to determine a branch address. Presenting E1h on PORTB will cause the program to jump to and execute the "program/verify" routine.

Note:	The Oscillator must not have 72 OSC clocks while the device $\overline{MCLR}$ is between V_{IL} and V_{IHH}.

All unused pins during programming are in hi-impedance state.

PORTB (RB pins) has internal weak pull-ups which are active during the programming mode. When the TEST pin is high, the Power-up timer (PWRT) and Oscillator Start-up Timers (OST) are disabled.

2.1 Program/Verify Mode

The program/verify mode is intended for full-feature programmers. This mode offers the following capabilities:

a) Load any arbitrary 16-bit address to start program and/or verify at that location.

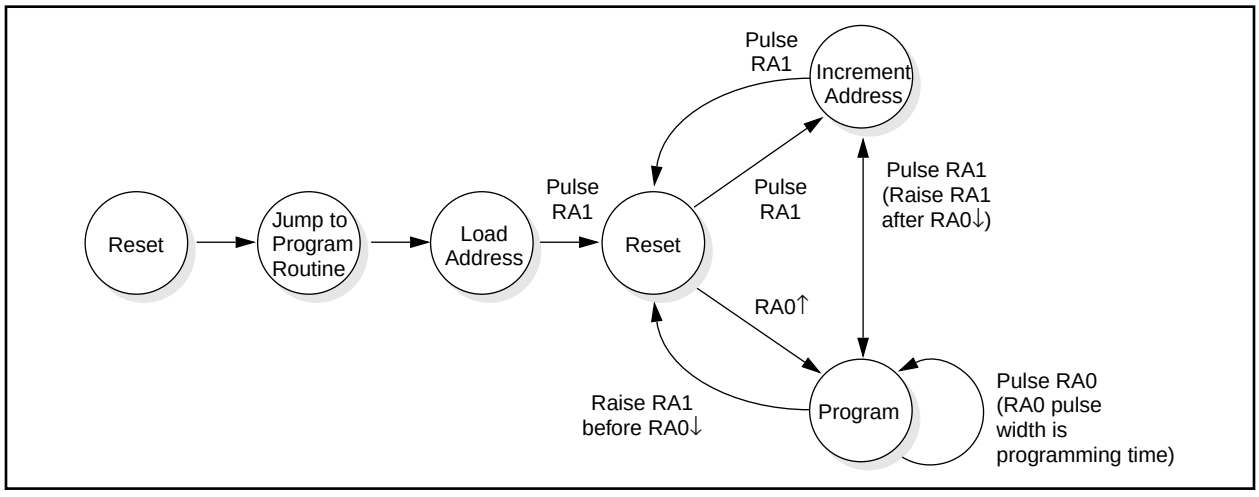
b) Increment address to program/verify the next location.

c) Allows arbitrary length programming pulse width.

d) Following a "verify" allows option to program the same location or increment and verify the next location.

e) Following a "program" allows options to program the same location again, verify the same location or to increment and verify the next location.

FIGURE 2-1: PROGRAMMING/VERIFY STATE DIAGRAM



2.1.1 LOADING NEW ADDRESS

The program allows new address to be loaded right out of reset. A 16-bit address is presented on ports B (high byte) and C (low byte) and the RA1 is pulsed ($0 \rightarrow 1$, then $1 \rightarrow 0$). The address is latched on the rising edge of RA1. See timing diagrams for details. After loading an address, the program automatically goes into a "verify cycle." To load a new address at any time, the PIC17C7XX must be reset and the programming mode re-entered.

2.1.2 VERIFY (OR READ) MODE

"Verify mode" can be entered from "Load address" mode, "program mode" or "verify mode." In verify mode pulsing RA1 will turn on PORTB and PORTC output drivers and output the 16-bit value from the current location. Pulsing RA1 again will increment location count and be ready for the next verify cycle. Pulsing RA0 will begin a program cycle.

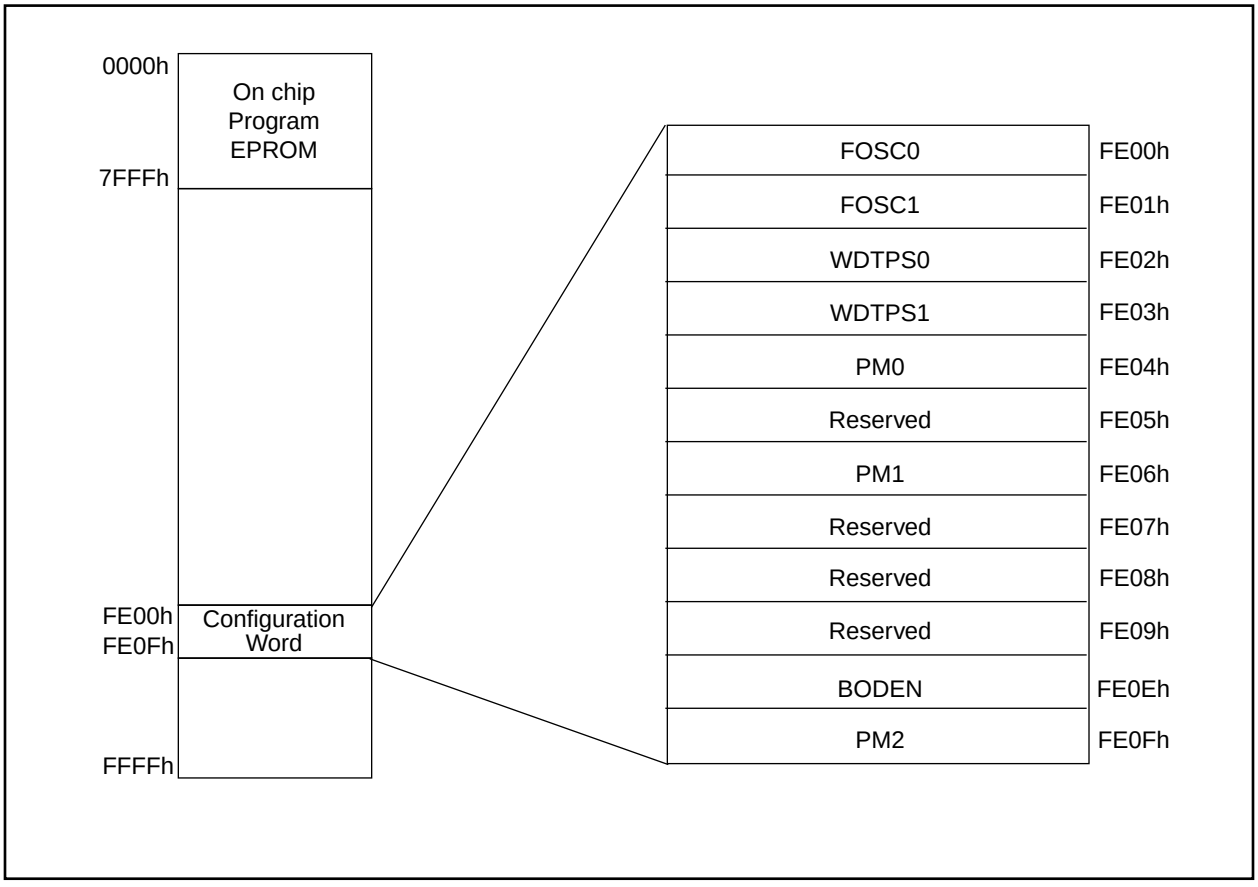
2.1.3 PROGRAM CYCLE

"Program cycle" is entered from "verify cycle" or program cycle" itself. After a verify, pulsing RA0 will begin a program cycle. 16-bit data must be presented on PORTB (high byte) and PORTC (low byte) before RA0 is raised.

The data is sampled 3 TCY cycles after the rising edge of RA0. Programming continues for the duration of RA0 pulse.

At the end of programming, the user can choose one of three different routes. If RA1 is kept low and RA0 is pulsed again, the same location will be programmed again. This is useful for applying over programming pulses. If RA1 is raised before RA0 falling edge, then a verify cycle is started without address increment. Raising RA1 after RA0 goes low will increment address and begin verify cycle on the next address.

FIGURE 2-2: PIC17C7XX PROGRAM MEMORY MAP



3.0 PARALLEL MODE PROGRAMMING SPECIFICATIONS

FIGURE 3-1: PROGRAMMING ROUTINE FLOWCHART

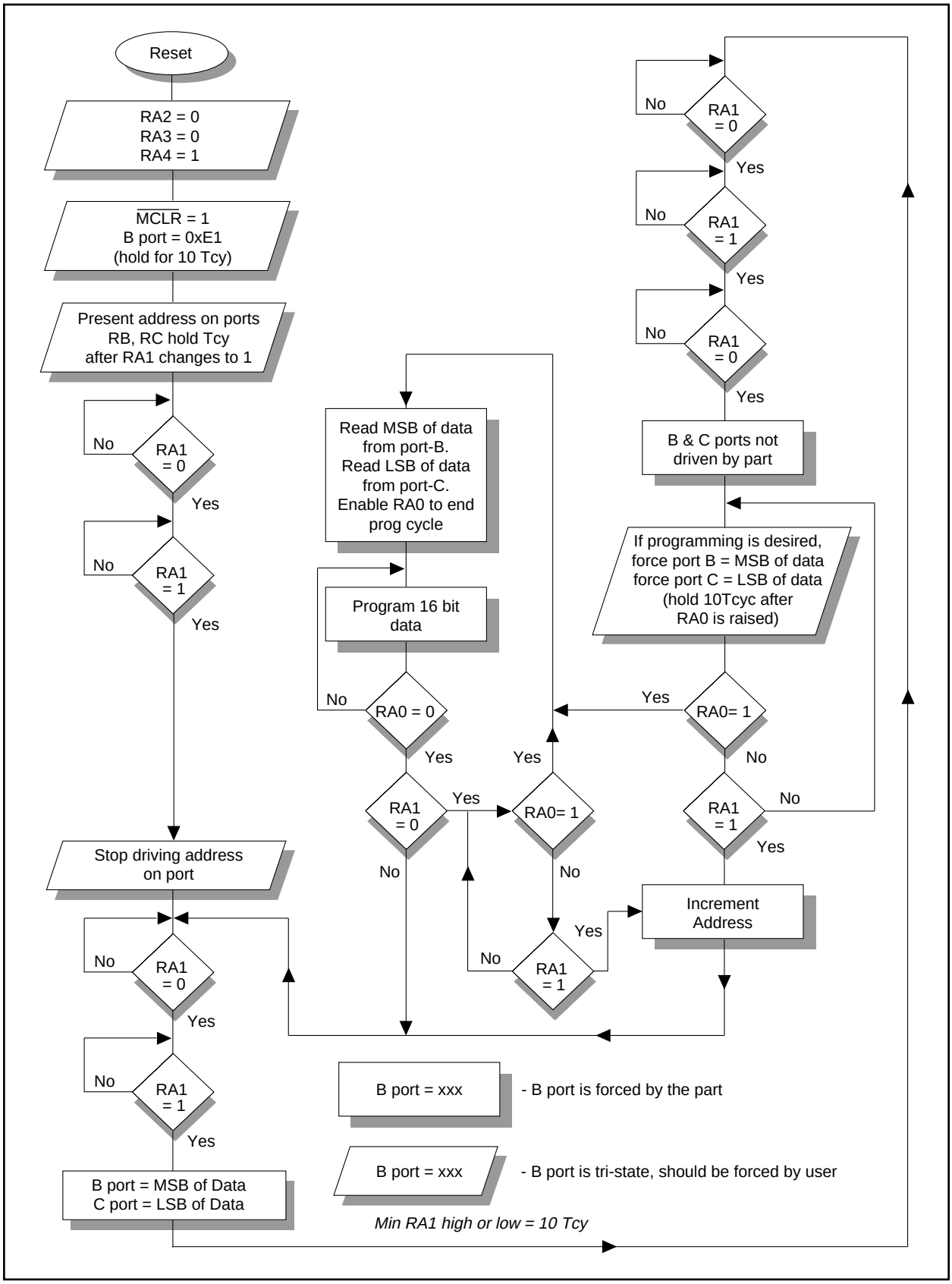


FIGURE 3-2: RECOMMENDED PROGRAMMING ALGORITHM FOR USER EPROM

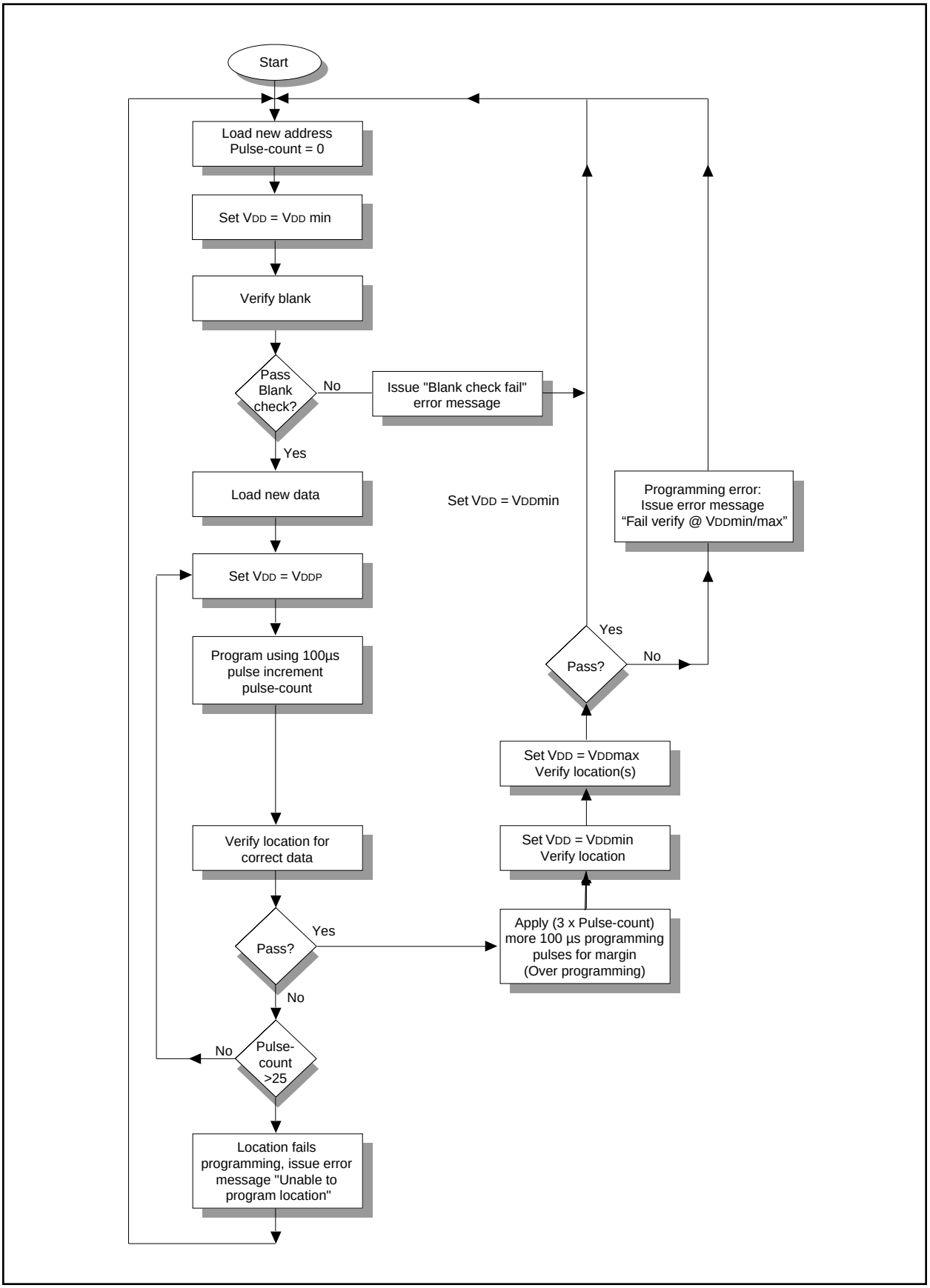
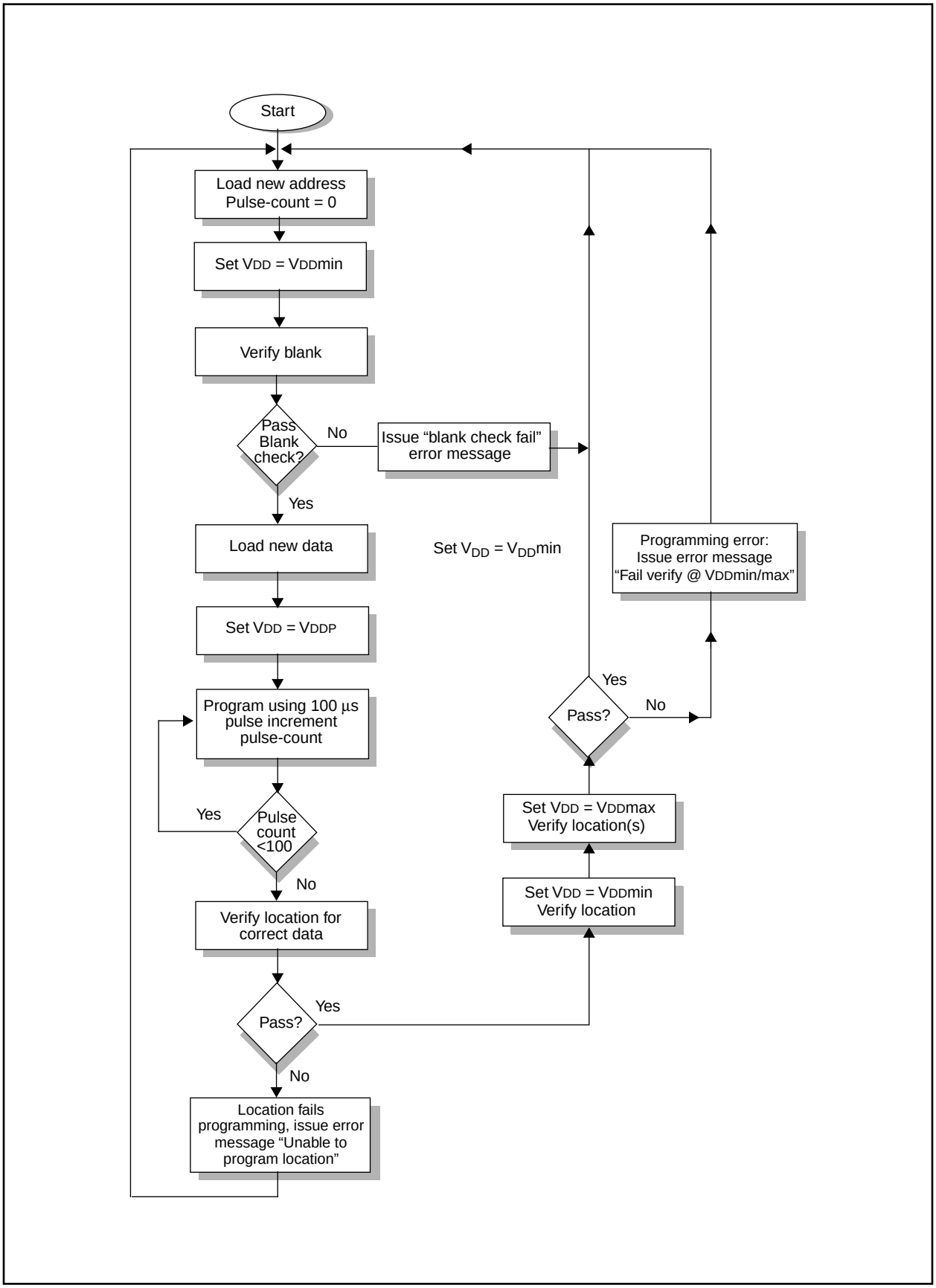


FIGURE 3-3: RECOMMENDED PROGRAMMING ALGORITHM FOR CONFIGURATION WORDS



4.0 SERIAL MODE PROGRAM ENTRY

4.1 Hardware Requirements

Certain design criteria must be taken into account for ISP. Seven pins are required for the interface. These are shown in Table 4-1.

4.2 Serial Program Mode Entry

To place the device into the serial programming test mode, two pins will need to be placed at VIHH. These are the TEST pin and the $\overline{MCLR}$/VPP pins. Also, the following sequence of events must occur:

1. The TEST pin is placed at VIHH.
2. The $\overline{MCLR}$/VPP pin is placed at VIHH.

There is a setup time between step 1 and step 2 that must be meet (See “Electrical Specifications for Serial Programming Mode” on page 22.)

After this sequence the Program Counter is pointing to Program Memory Address 0xFF60. This location is in the Boot ROM. The code initializes the USART/SCI so that it can receive commands. For this the device must be clocked. The device clock source in this mode is the RA1/T0CKI pin. Once the USART/SCI has been initialized, commands may be received. The flow is show in these 3 steps:

1. The device clock source starts.
2. Wait 80 device clocks for Boot ROM code to configure the USART/SCI.
3. Commands may be sent now.

Table 4-1: ISP Interface Pins

Name	During Programming		
	Function	Type	Description
RA4/RX/DT	DT	I/O	Serial Data
RA5/TX/CK	CK	I	Serial Clock
RA1/T0CKI	OSCI	I	Device Clock Source
TEST	TEST	I	Test mode selection control input. Force to VIHH,
$\overline{MCLR}$/VPP	$\overline{MCLR}$/VPP	P	Master Clear reset and Device Programming Voltage
VDD	VDD	P	Positive supply for logic and I/O pins
VSS	VSS	P	Ground reference for logic and I/O pins

4.3 Software Commands

This feature is similar to that of the PIC16CXXX midrange family, but the programming commands have been implemented in the device Boot ROM. The Boot ROM is located in the program memory from 0xFF60 to 0xFFFF. The ISP mode is entered when the TEST pin has a VIHH voltage applied. Once in ISP mode, the USART/SCI module is configured as a synchronous slave receiver, and the device waits for a command to be received. The ISP firmware recognizes eight commands. These are shown in Table 4-1.

TABLE 4-1: ISP COMMANDS

Command	Value
RESET PROGRAM MEMORY POINTER	0000 0000
LOAD DATA	0000 0010
READ DATA	0000 0100
INCREMENT ADDRSS	0000 0110
BEGIN PROGRAMMING	0000 1000
LOAD ADDRESS	0000 1010
READ ADDRESS	0000 1100
END PROGRAMMING	0000 1110

4.3.1 RESET PROGRAM MEMORY POINTER

This is used to clear the address pointer to the Program Memory. This ensures that the pointer is at a known state as well as pointing to the first location in program memory.

4.3.2 INCREMENT ADDRESS

This is used to increment the address pointer to the Program Memory. This is used after the current location has been programmed (or read).

FIGURE 4-1: RESET ADDRESS POINTER COMMAND (PROGRAM/VERIFY)

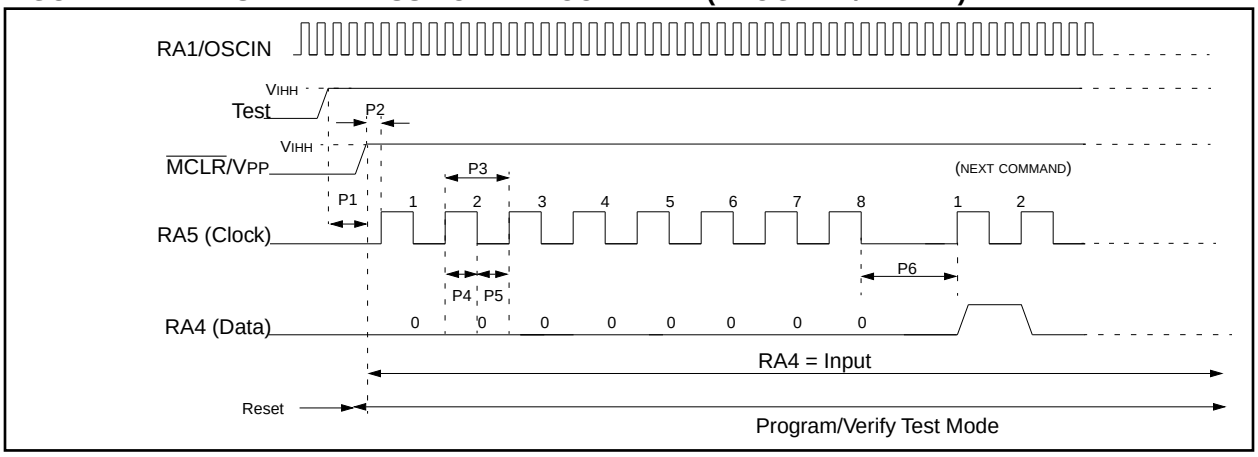
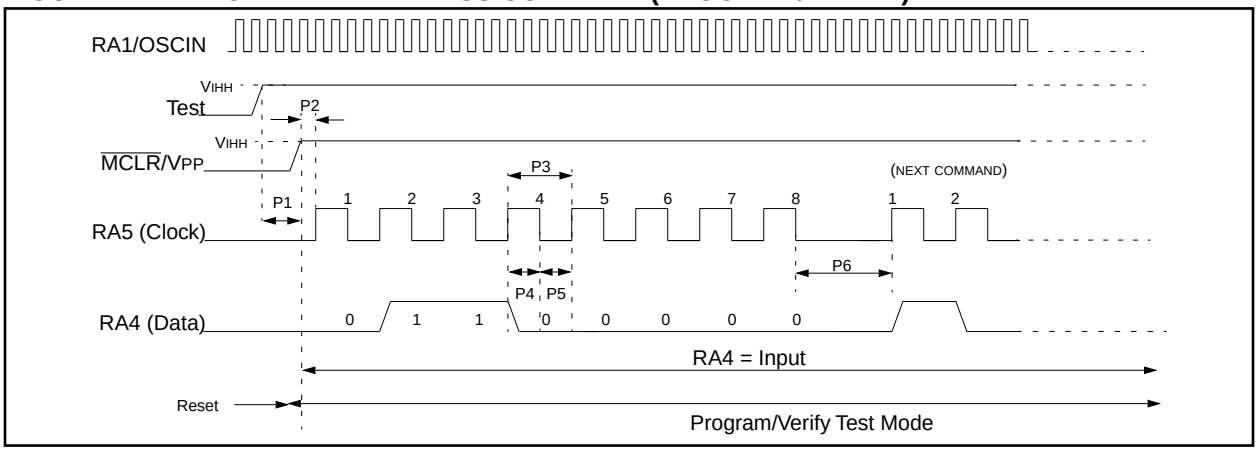


FIGURE 4-2: INCREMENT ADDRESS COMMAND (PROGRAM/VERIFY)



4.3.3 LOAD ADDRESS

This is used to load the address pointer to the Program Memory with a specific 16-bit value. This is useful when a specific range of locations are to be accessed.

4.3.4 READ ADDRESS

This is used so that the current address in the Program Memory pointer can be determined. This can be used to increase the robustness of the ISP programming (ensure that the Program Memory pointers are still in sync).

FIGURE 4-3: LOAD ADDRESS COMMAND

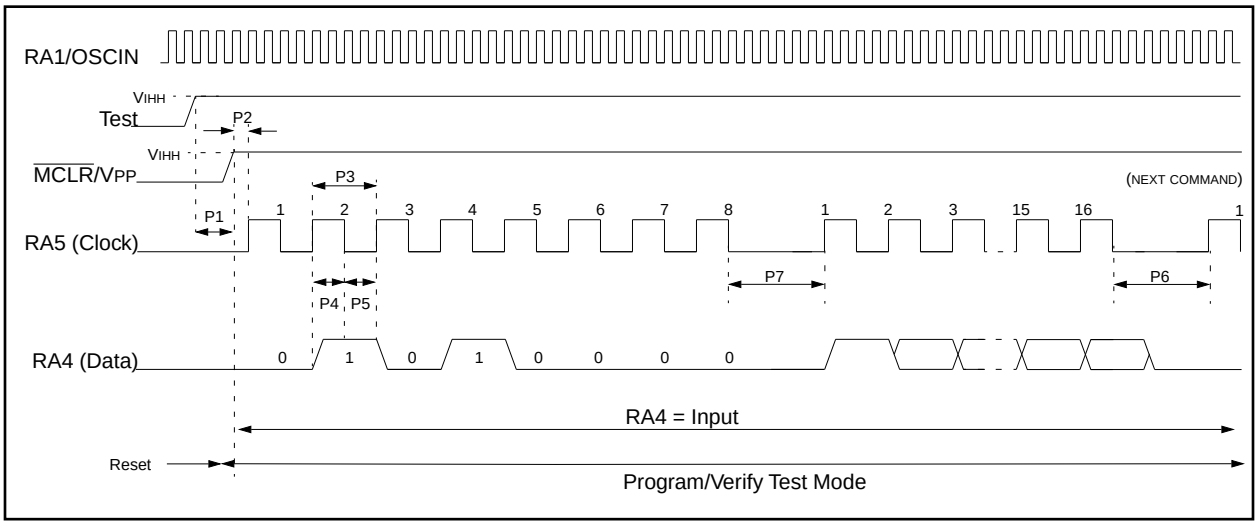
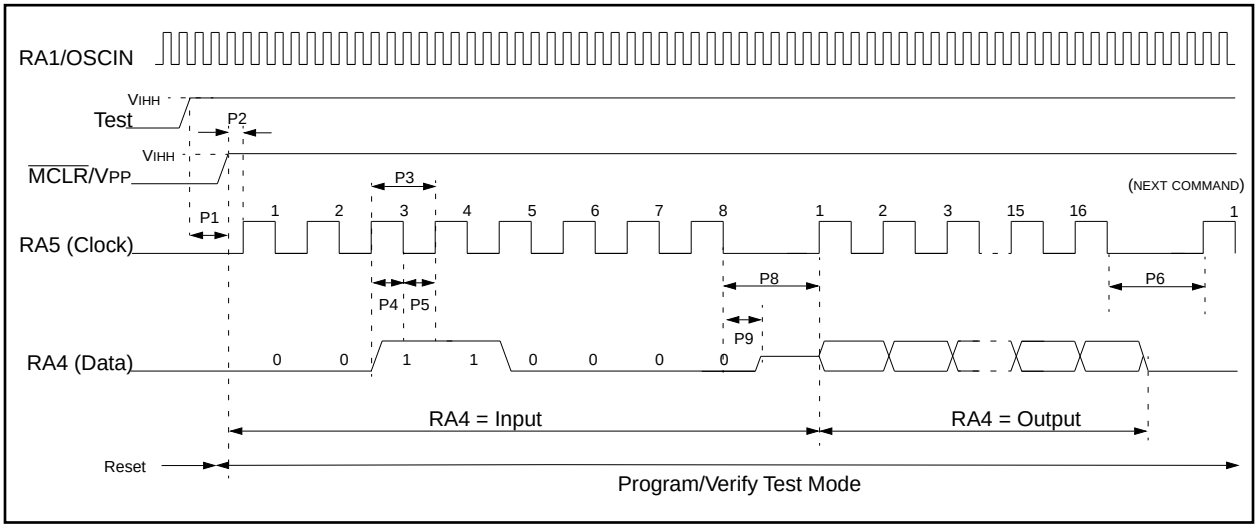


FIGURE 4-4: READ ADDRESS COMMAND



4.3.5 LOAD DATA

This is used to load the 16-bit data that is to be programmed into the Program Memory location. The Program Memory address may be modified after the data is loaded. This data will not be programmed until a BEGIN PROGRAMMING command is executed.

4.3.6 READ DATA

This is used to read the data in Program Memory that is pointed to by the current address pointer. This is useful for doing a verify of the programming cycle and can be used to determine the number for programming cycles that are required for the 3X overprogramming.

FIGURE 4-5: LOAD DATA COMMAND

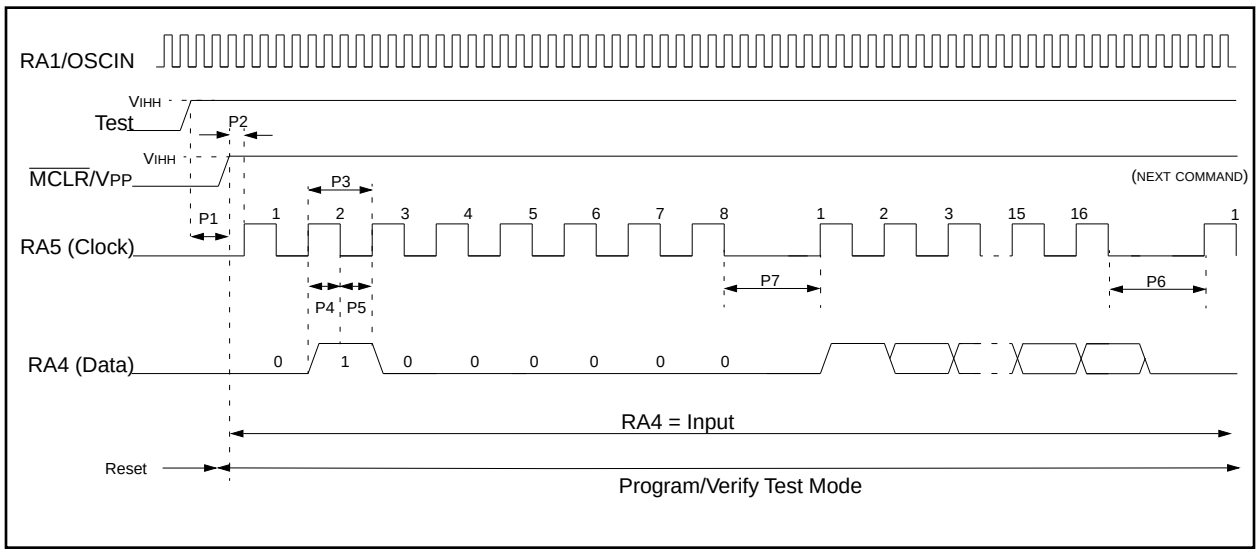
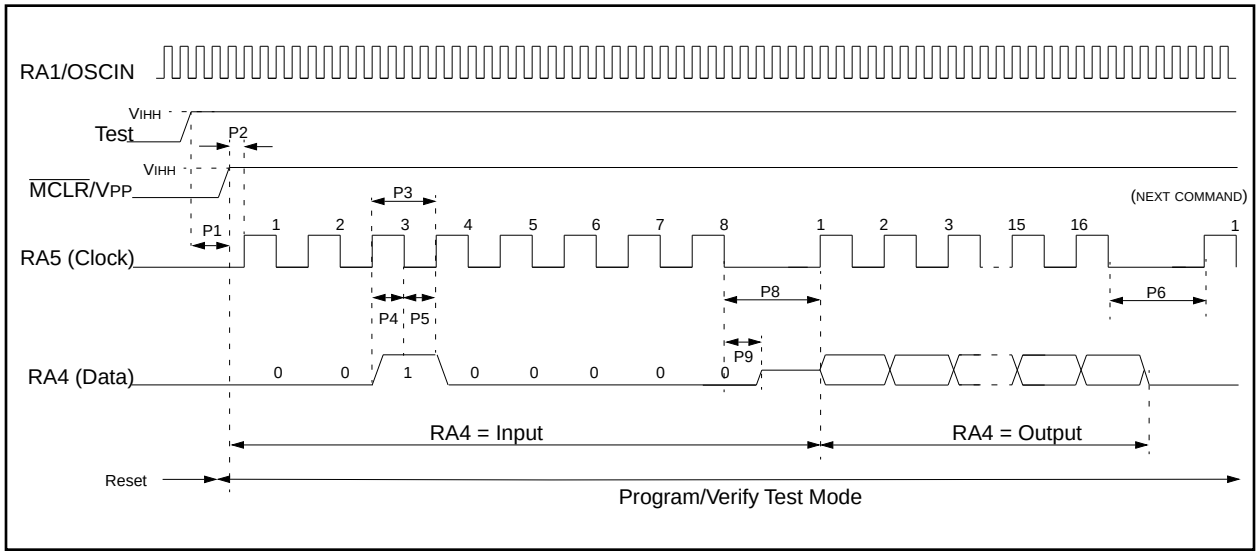


FIGURE 4-6: READ DATA COMMAND



4.3.7 BEGIN PROGRAMMING

This is used to program the current 16-bit data (last data sent with LOAD DATA Command) into the Program Memory at the address specified by the current address pointer. The programming cycle time is specified by specification P10. After this time has elasped, any command must be sent, which wakes the processor from the Long Write cycle. This command will be the next executed command.

4.3.8 3X OVERPROGRAMMING

Once a location has been both programmed and verified over a range of voltages, 3X overprogramming should be applied. In other words, apply three times the number of programming pulses that were required to program a location in memory, to ensure a solid programming margin.

This means that every location will be programmed a minimum of 4 times (1 + 3X overprogramming).

FIGURE 4-7: BEGIN PROGRAMMING COMMAND (PROGRAM)

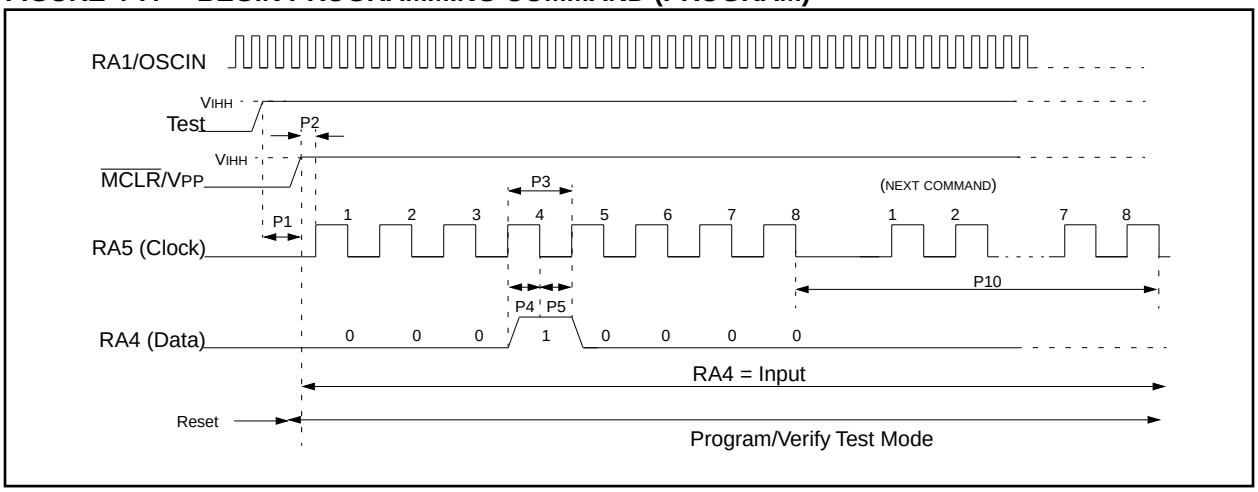
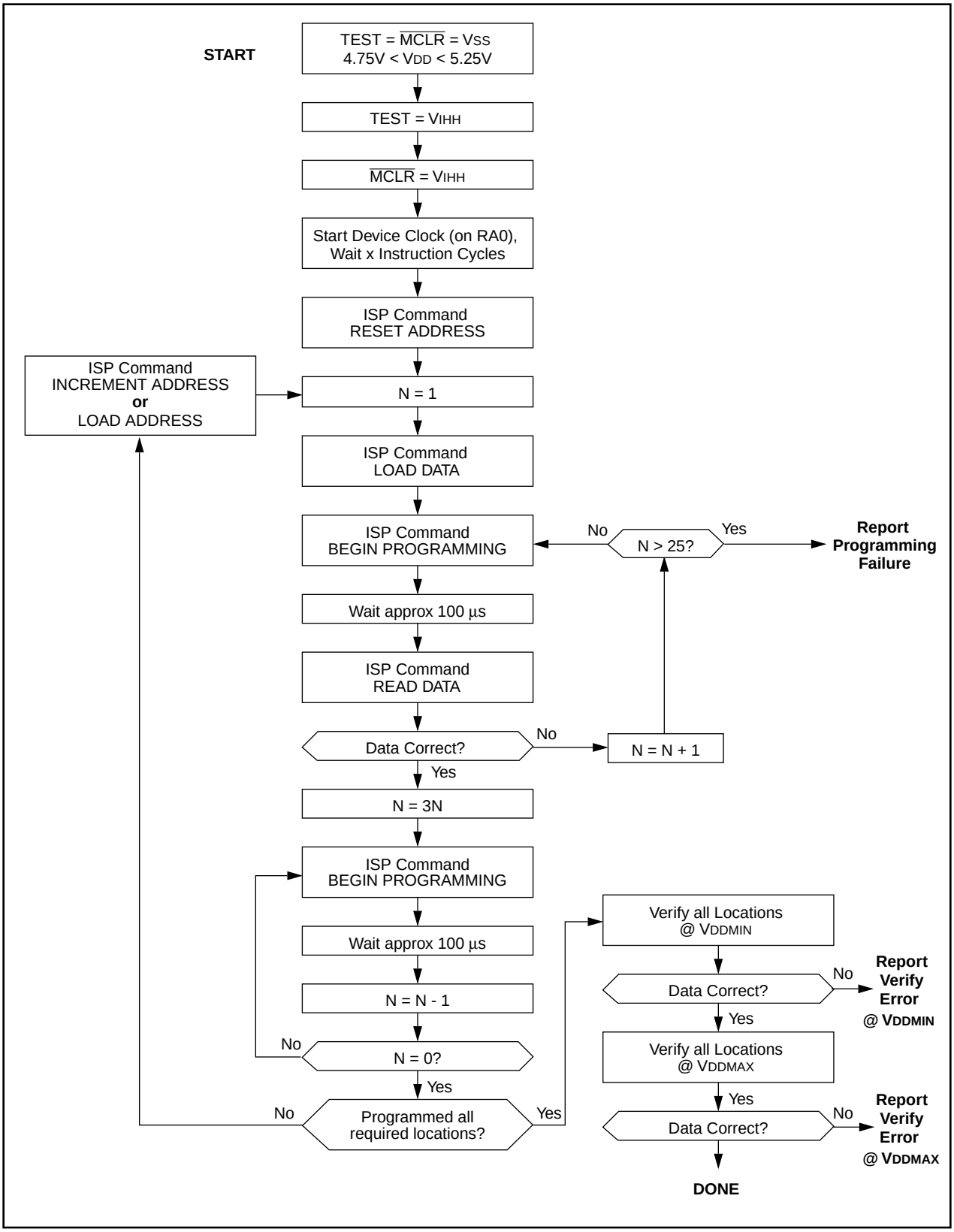


FIGURE 4-8: RECOMMENDED PROGRAMMING FLOWCHART



5.0 CONFIGURATION WORD

Configuration bits are mapped into program memory. Each bit is assigned one memory location. In erased condition, a bit will read as '1'. To program a bit, the user needs to write to the memory address. The data is immaterial; the very act of writing will program the bit. The configuration word locations are shown in Table 5-3. **The programmer should not program the reserved locations to avoid unpredictable results and to be compatible with future variations of the PIC17C4X. It is also mandatory that configuration locations are programmed in the strict order starting from the first location (0xFE00) and ending with the last (0xFE0F). Unpredictable results may occur if the sequence is violated.**

5.1 Reading Configuration Word

The PIC17C5X has seven configuration locations (Table 5-1). These locations can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. Any write to a configuration location, regardless of the data, will program that configuration bit. Reading any configuration location between 0xFE00 and 0xFE07 will place the low byte of the configuration word (Table 5-2) into DAD<7:0> (PORTC). DAD<15:8> (PORTD) will be set to 0xFF. Reading a configuration location between 0xFE08 and 0xFE0F will place the high byte of the configuration word into DAD<7:0> (PORTC). DAD<15:8> (PORTD) will be set to 0xFF.

TABLE 5-1: CONFIGURATION BIT PROGRAMMING LOCATIONS

Bit	Address
FOSC0	0xFE00
FOSC1	0xFE01
WDTPS0	0xFE02
WDTPS1	0xFE03
PM0	0xFE04
PM1	0xFE06
BODEN	0xFE0E
PM2	0xFE0F

TABLE 5-2: READ MAPPING OF CONFIGURATION BITS

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	—	PM1	—	PM0	WDTPS1	WDTPS0	FOSC1	FOSC0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	PM2	BODEN	—	—	—	—	—	—

—=Unused

PM<2:0>, Processor Mode Select bits

111 = Microprocessor mode
110 = Microcontroller mode
101 = Extended Microcontroller mode
000 = Code protected microcontroller mode

BODEN, Brown-out Detect Enable

1 = Brown-out Detect Circuitry enabled
0 = Brown-out Detect Circuitry disabled

WDTPS1:WDTPS0, WDT Prescaler Select bits.

11 = WDT enabled, postscaler = 0
10 = WDT enabled, postscaler = 256
01 = WDT enabled, postscaler = 64
00 = WDT disabled, 16-bit overflow timer

FOSC1:FOSC0, Oscillator Select bits

11 = EC oscillator
10 = XT oscillator
01 = RC oscillator
00 = LF oscillator

5.2 Embedding Configuration Word Information in the Hex File

To allow portability of code, a PIC17C7XX programmer is required to read the configuration word locations from the hex file when loading the hex file. If the configuration word information was not present in the hex file, then a simple warning message may be issued. Similarly, while saving a hex file, all configuration word information must be included. An option to not include the configuration word information may be provided. When embedding configuration word information in the hex file, it should be to address FE00h.

Microchip Technology Inc. feels strongly that this feature is important for the benefit of the end customer.

TABLE 5-3: CONFIGURATION WORD

PIC17C752

To code protect:

- Protect all memory `0XXXXXXX0X0XXXX`

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFE00)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
All memory	Read Scrambled, Write Disabled*	Read Unscrambled, Write Enabled

PIC17C756

To code protect:

- Protect all memory `0XXXXXXX0X0XXXX`

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFE00)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
All memory	Read Scrambled, Write Disabled*	Read Unscrambled, Write Enabled

Legend: X = Don't care

*Write to on-chip EPROM memory is disabled. The only way these locations can be programmed is if a TABLWT instruction is issued from an "on-chip" program memory space to program an on-chip memory location.

5.3 CHECKSUM COMPUTATION

The checksum is calculated by summing the following:

- The contents of all program memory locations
- The configuration word, appropriately masked
- Masked ID locations (when applicable)

The least significant 16 bits of this sum is the checksum.

Table 5-4 describes how to calculate the checksum for each device. Note that the checksum calculation differs depending on the code protect setting. Since the program memory locations read out differently, depending on the code protect setting, the table describes how to manipulate the actual program memory values to simulate the values that would be read from a protected device. When calculating a checksum by reading a device, the entire program memory can simply be read and summed. The configuration word and ID locations can always be read.

Note:	Some older devices have an additional value added in the checksum. This is to maintain compatibility with older device programmer checksums.

TABLE 5-4: CHECKSUM COMPUTATION

Device	Code Protect	Checksum*	Blank Value	0xC0DE at 0 and max address
PIC17C752	MP mode	SUM[0x0000:0x1FFF] + (CONFIG & 0xC05F)	0xA05F	0x221D
	MC mode	SUM[0x0000:0x1FFF] + (CONFIG & 0xC05F)	0xA04F	0x220D
	EMC mode	SUM[0x0000:0x1FFF] + (CONFIG & 0xC05F)	0xA01F	0x21DD
	PMC mode	SUM_XNOR8[0x0000:0x1FFF] + (CONFIG & 0xC05F)	0x200F	0xE3D3
PIC17C756	MP mode	SUM[0x0000:0x3FFF] + (CONFIG & 0xC05F)	0x805F	0x021D
	MC mode	SUM[0x0000:0x3FFF] + (CONFIG & 0xC05F)	0x804F	0x020D
	EMC mode	SUM[0x0000:0x3FFF] + (CONFIG & 0xC05F)	0x801F	0x01DD
	PMC mode	SUM_XNOR8[0x0000:0x3FFF] + (CONFIG & 0xC05F)	0x000F	0xC3D3

Legend: CFGW = Configuration Word
SUM[a:b] = [Sum of locations a to b inclusive]
SUM_XNOR8(a:b) = [Sum of 8-bit wide XNOR copied into upper and lower byte, of locations a to b inclusive]
*Checksum = [Sum of all the individual expressions] **MODULO** [0xFFFF]
+ = Addition
& = Bitwise AND

6.0 PARALLEL MODE AC/DC CHARACTERISTICS AND TIMING REQUIREMENTS FOR PROGRAM/VERIFY TEST MODE

Standard Operating Conditions

Operating Temperature: +10°C ≤ TA ≤ +70°C, unless otherwise stated, (25°C is recommended)
Operating Voltage: 4.5V ≤ VDD ≤ 5.25V, unless otherwise stated.

Parameter No.	Sym.	Characteristic	Min.	Typ.	Max.	Units	Conditions/Comments
PD1	VDDP	Supply voltage during programming	4.75	5.0	5.25	V	
PD2	IDDP	Supply current during programming	—	—	50	mA	Freq = 10MHz, VDD = 5.5V Note 3
PD3	VDDV	Supply voltage during verify	VDD min.	—	VDD max.	V	Note 2
PD4	VPP	Voltage on VPP/$\overline{MCLR}$ pin during programming	12.75	—	13.25	V	Note 1
PD6	IPP	Programming current on VPP/$\overline{MCLR}$ pin	—	25	50	mA	Note 3
P1	FOSCP	Osc/clockin frequency during programming	4	—	10	MHz	
P2	TCY	Instruction cycle	1	—	0.4	µs	TCY = 4/FOSCP
P3	TIRV2TSH	RA0, RA1, RA2, RA3, RA4 setup before TEST↑	1	—	—	µs	
P4	TTSH2MCH	TEST↑ to $\overline{MCLR}$↑	1	—	—	µs	
P5	TBCV2IRH	RC7:RC0, RB7:RB0 valid to RA1 or RA0↑:Address/ Data input setup time	0	—	—	µs	?PORTD vs. PORTB? AL?
P6	TIRH2BCL	RA1 or RA0↑ to RB7:RB0, RC7:RC0 invalid; Address data hold time;	10 TCY	—	—	µs	?PORTD vs. PORTB? AL?
P7	T0CKIL2RBCZ	RT↓ to RB7:RB0, RC7:RC0 hi-impedance	—	—	8TCY		?PORTD vs. PORTB? AL?
P8	T0CKIH2BCV	RA1↑ to data out valid	—	—	10 TCY		
P9	TPROG	Programming pulse width	10	100	1000	µs	
P10	TIRH2IRL	RA0, RA1 high pulse width	10 TCY	—	—	µs	
P11	TIRL2IRH	RA0, RA1 low pulse width	10 TCY	—	—	µs	
P12	T0CKIV2INL	RA1↑ before INT↓ (to go from prog cycle to verify w/o increment)	0	—	—	µs	
P13	TINL2RTL	RA1 valid after RA0 (to select increment or no increment going from program to verify cycle	10 TCY	—	—	µs	
P14	TVPPS	VPP setup time before RA0↑	100	—	—	µs	Note 1
P15	TVPPH	VPP hold time after INT↓	0	—	—	µs	Note 1
P16	TVDV2TSH	VDD stable to TEST↑	10	—	—	ms	
P17	TRBV2MCH	RB input (E1h) valid to VPP/ $\overline{MCLR}$↑	0	—	—	µs	
P18	TMCH2RBI	RB input (E1h) hold after VPP/$\overline{MCLR}$↑	10TCY	—	—	ns	
P19	TVPL2VDL	VDD power down after VPP power down	10	—	—	ms	

Note 1: VPP/$\overline{MCLR}$ pin must only be equal to or greater than VDD at times other than programming.
2: Program must be verified at the minimum and maximum VDD limits for the part.
3: These parameters are for design guidance only and are not tested nor characterized.

FIGURE 6-1: PARALLEL MODE PROGRAMMING AND VERIFY TIMINGS I

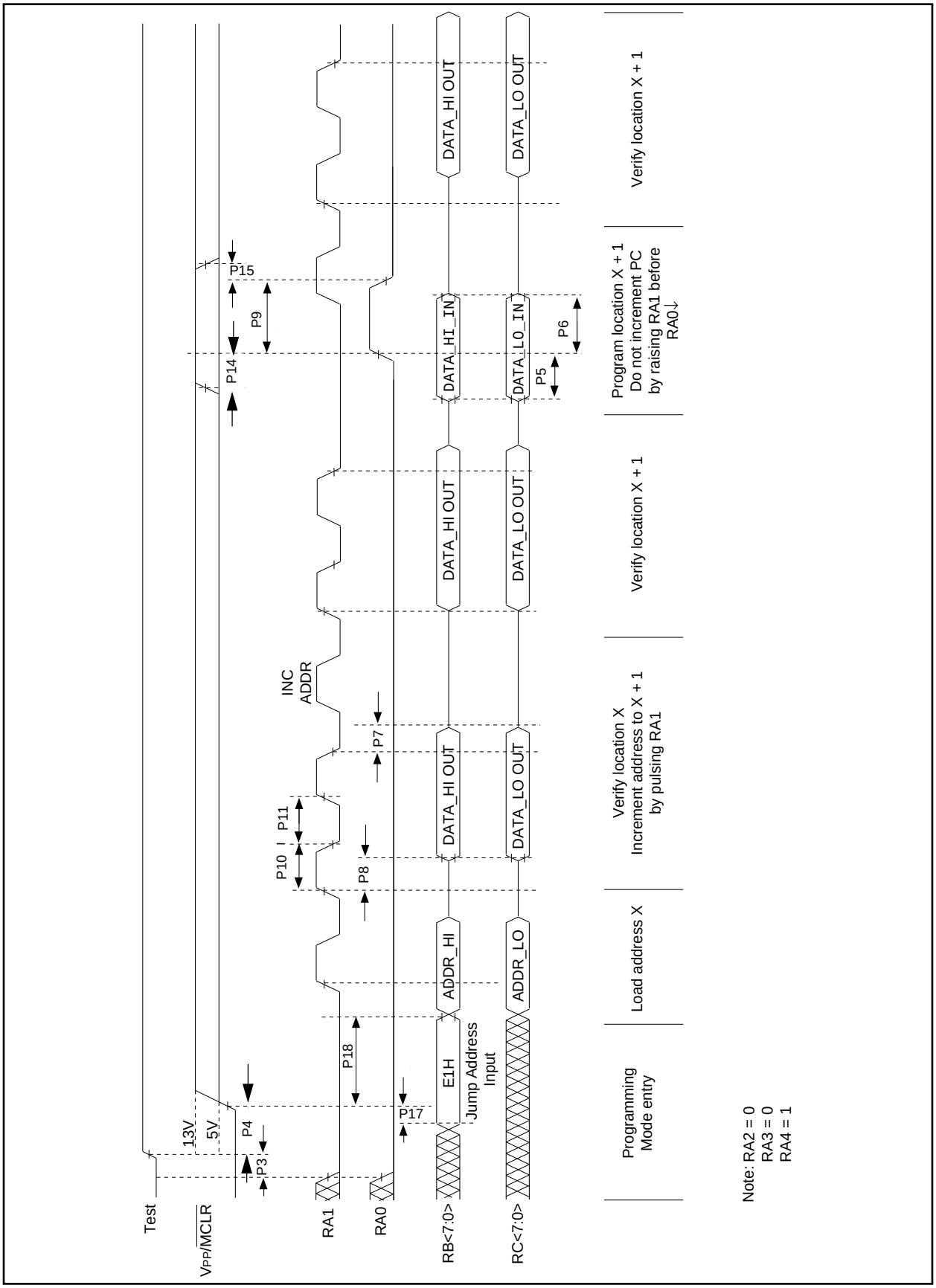


FIGURE 6-2: PARALLEL MODE PROGRAMMING AND VERIFY TIMINGS II

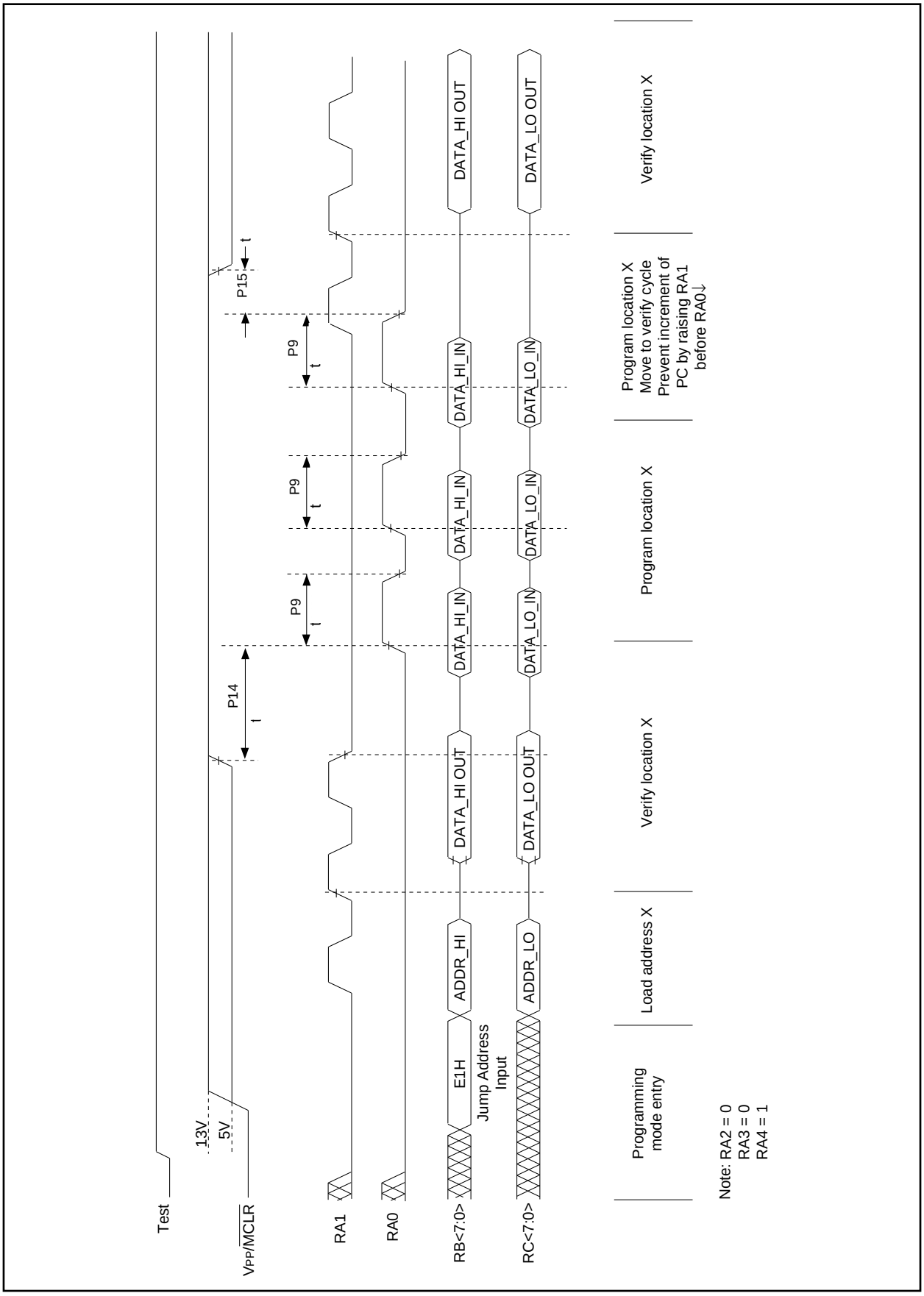


FIGURE 6-3: PARALLEL MODE PROGRAMMING AND VERIFY TIMINGS III

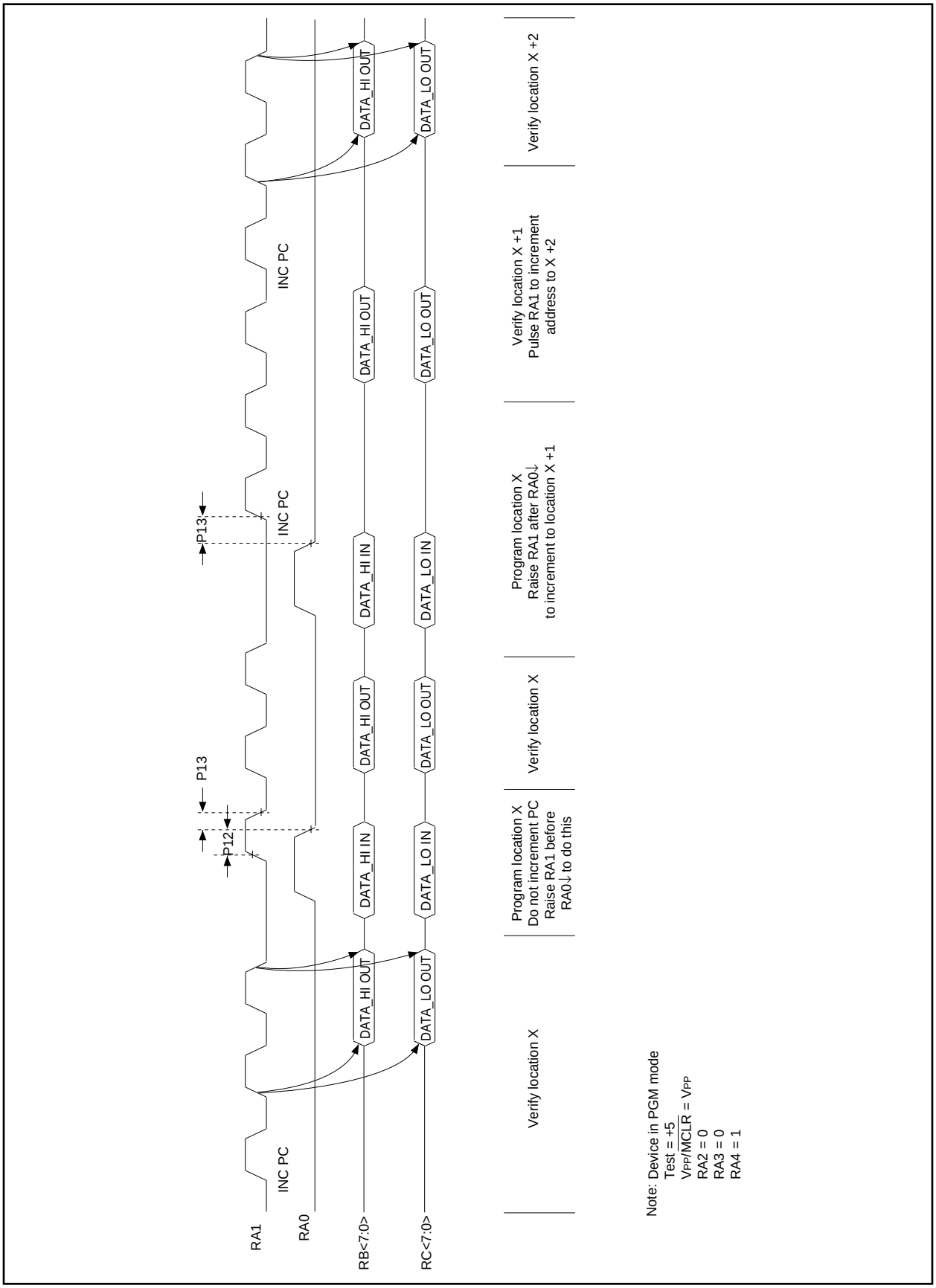
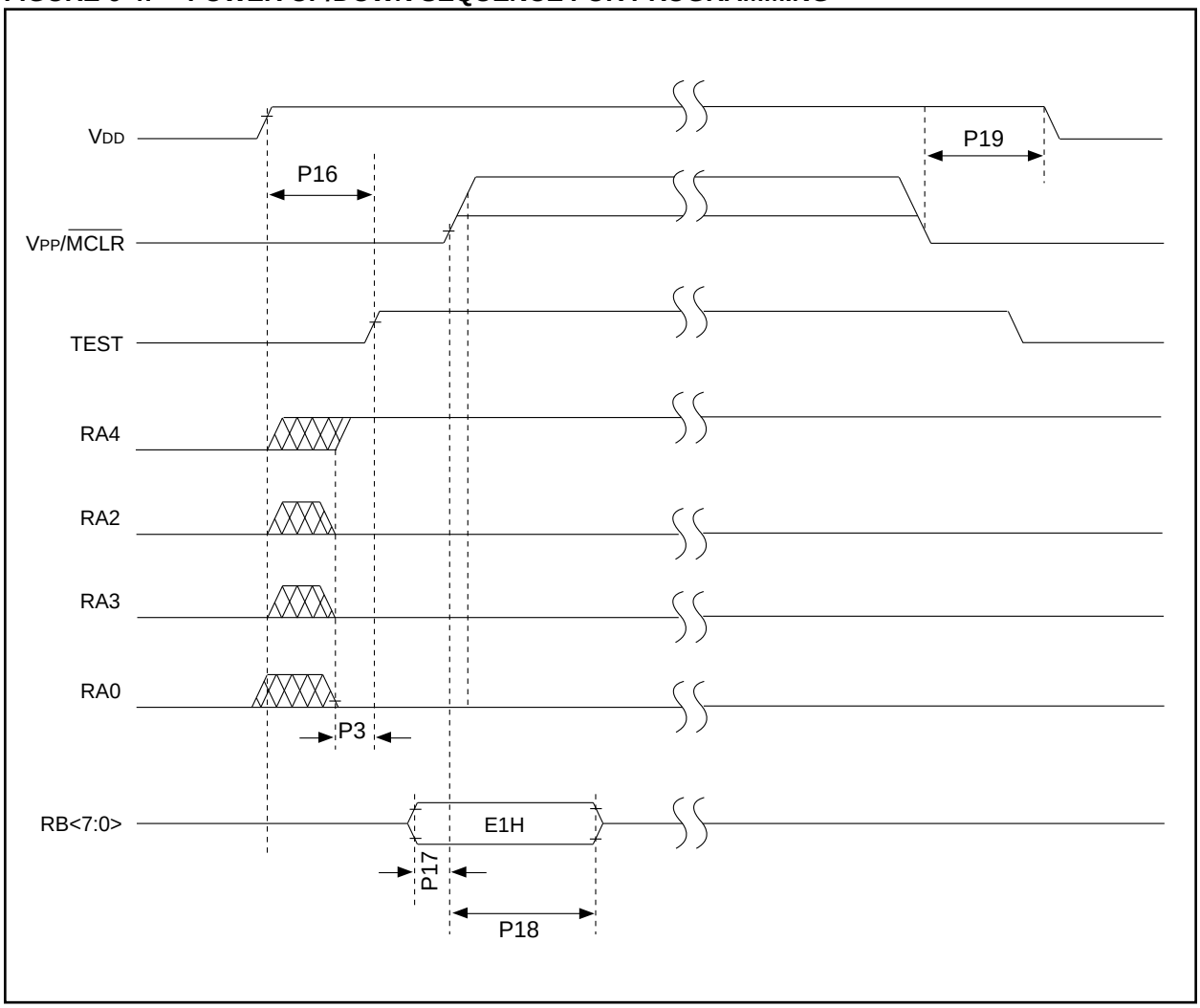


FIGURE 6-4: POWER-UP/DOWN SEQUENCE FOR PROGRAMMING



7.0 ELECTRICAL SPECIFICATIONS FOR SERIAL PROGRAMMING MODE

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	VIHH	Programming Voltage on VPP/ $\overline{MCLR}$ pin and TEST pin.	12.5	—	13.5	V	
	IPP	Programming current on $\overline{MCLR}$ pin	—	25	50	mA	
	FOSC	Input OSC frequency on RA1	—	—	8	MHz	
	TCY	Instruction Cycle Time	—	4/FOSC	—		
P1	TVH2VH	Setup time between TEST = VIHH and $\overline{MCLR}$ = VIHH	1	—	—	µs	
P2	TSER	Serial setup time	20	—	—	TCY	
P3	TSCLK	Serial Clock period	1	—	—	TCY	
P4	TSET1	Input Data Setup Time to serial clock ↓	15	—	—	ns	
P5	THLD1	Input Data Hold Time from serial clock ↓	15	—	—	ns	
P6	TDLY1	Delay between last clock ↓ to first clock ↑ of next command	20	—	—	TCY	
P7	TDLY2	Delay between last clock ↓ of command byte to first clock ↑ of read of data word	20	—	—	TCY	
P8	TDLY3	Delay between last clock ↓ of command byte to first clock ↑ of write of data word	30	—	—	TCY	
P9	TDLY4	Data input not driven to next clock input	1	—	—	TCY	
P10	TDLY5	Delay between last begin programming clock ↓ to last clock ↓ of next command (minimum programming time)	100	—	—	µs	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 7-1: RESET ADDRESS POINTER COMMAND (PROGRAM/VERIFY)

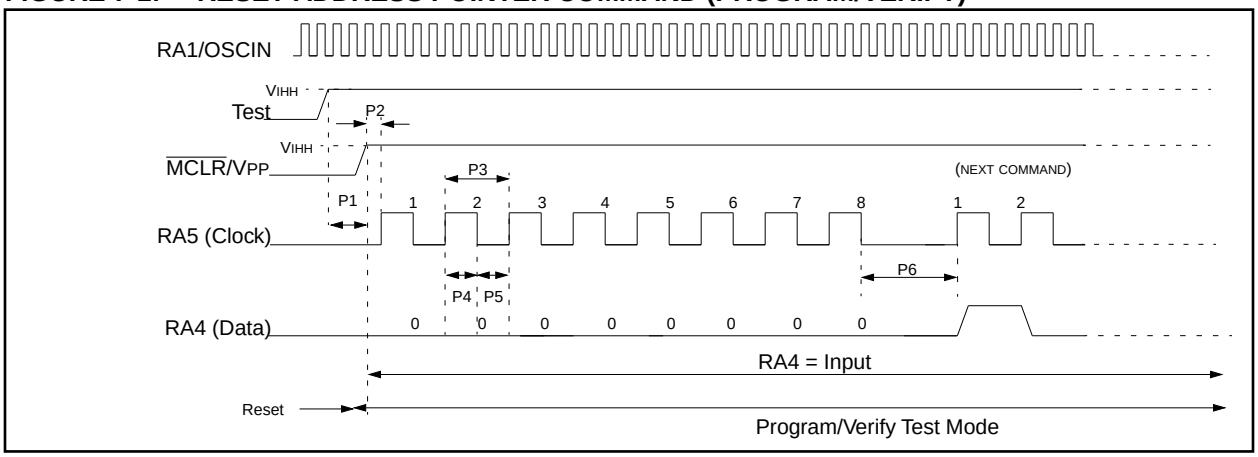


FIGURE 7-2: INCREMENT ADDRESS COMMAND (PROGRAM/VERIFY)

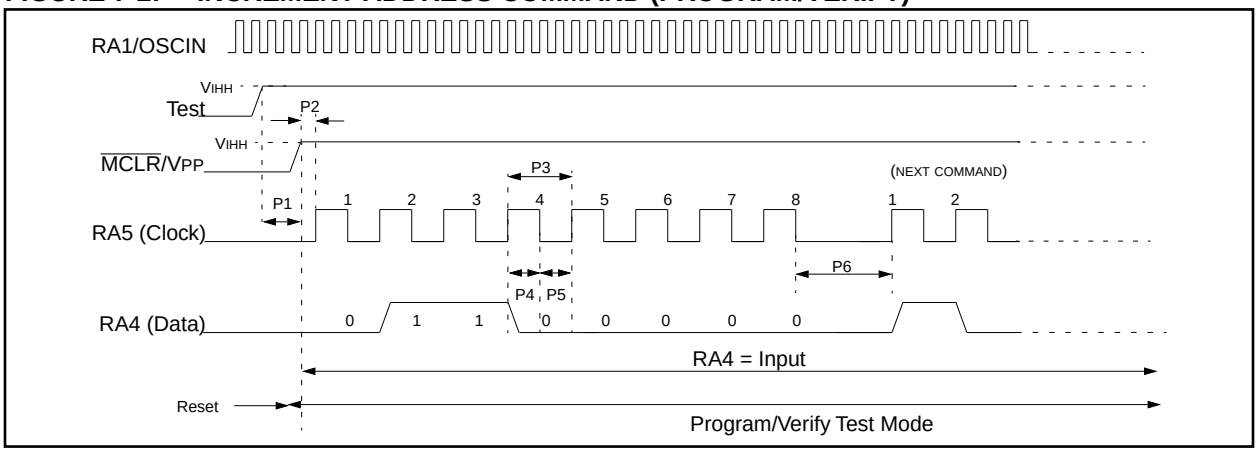


FIGURE 7-3: LOAD ADDRESS COMMAND

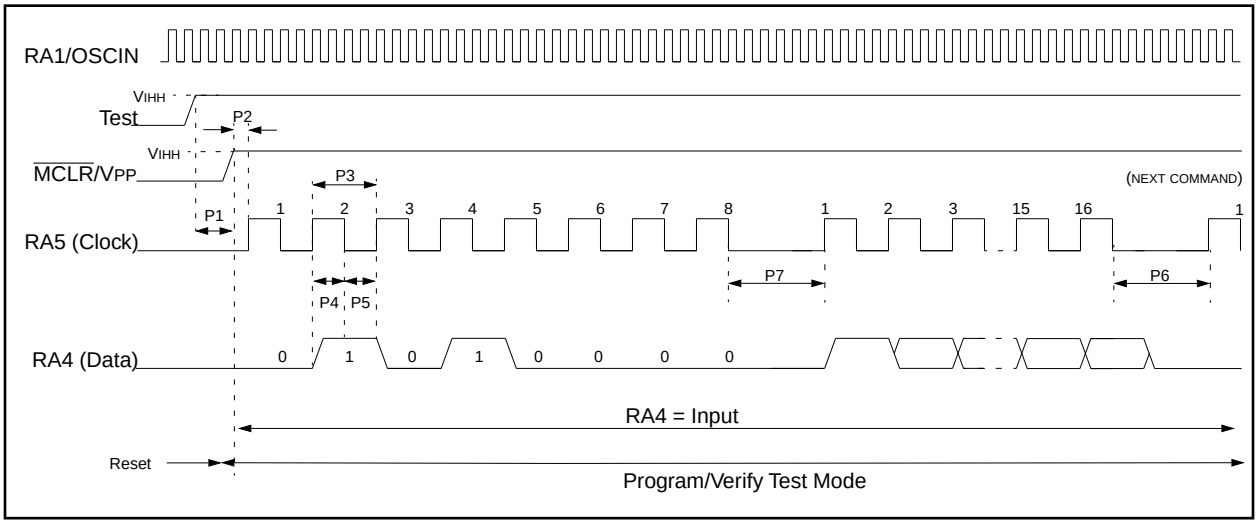


FIGURE 7-4: READ ADDRESS COMMAND

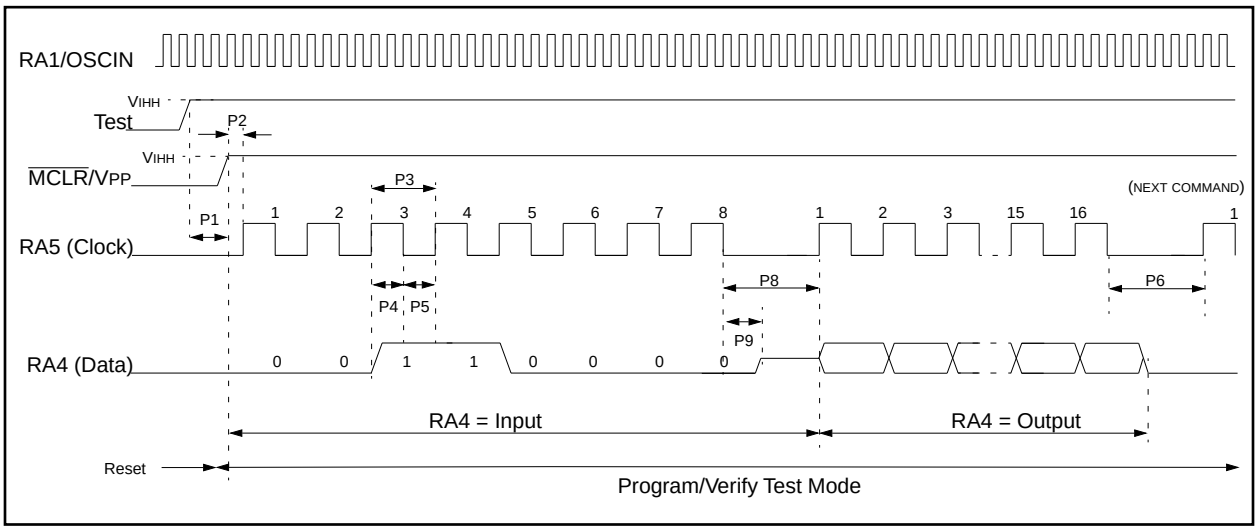


FIGURE 7-5: LOAD DATA COMMAND

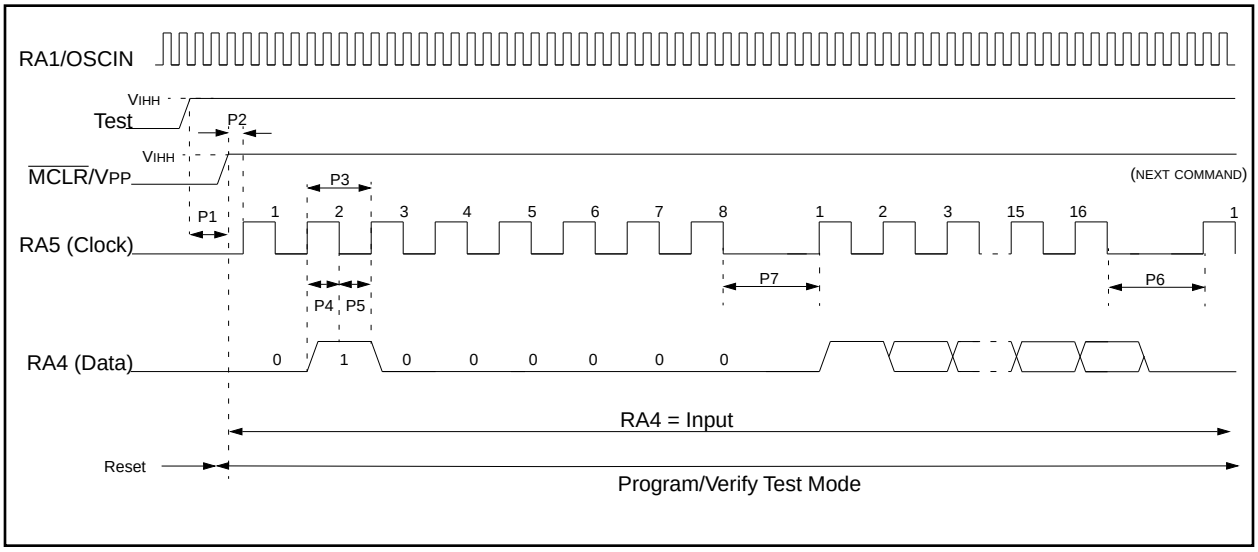


FIGURE 7-6: READ DATA COMMAND

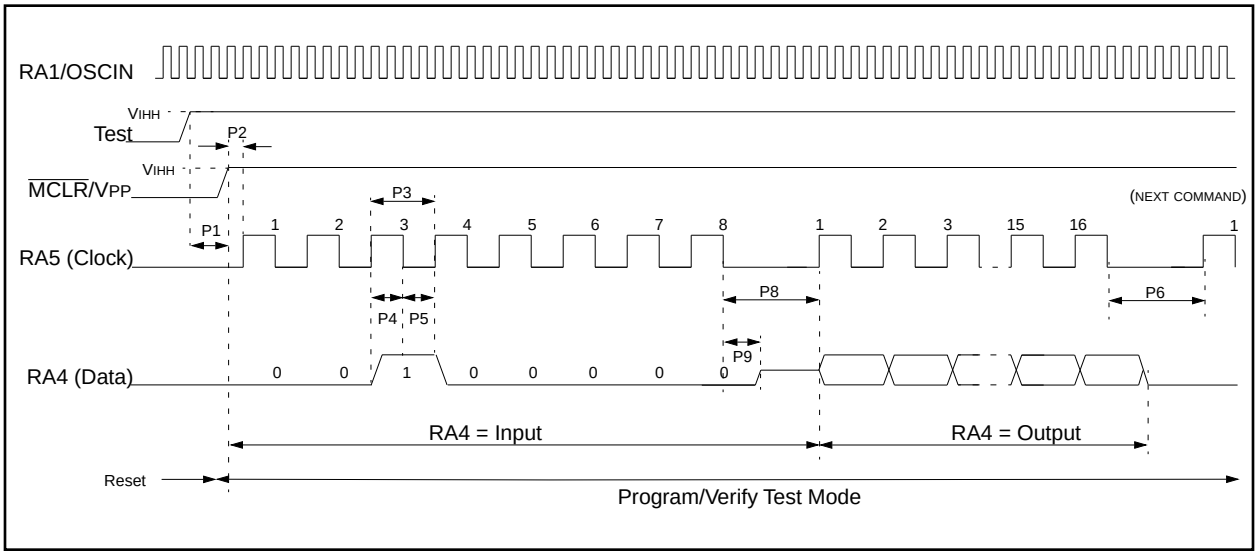
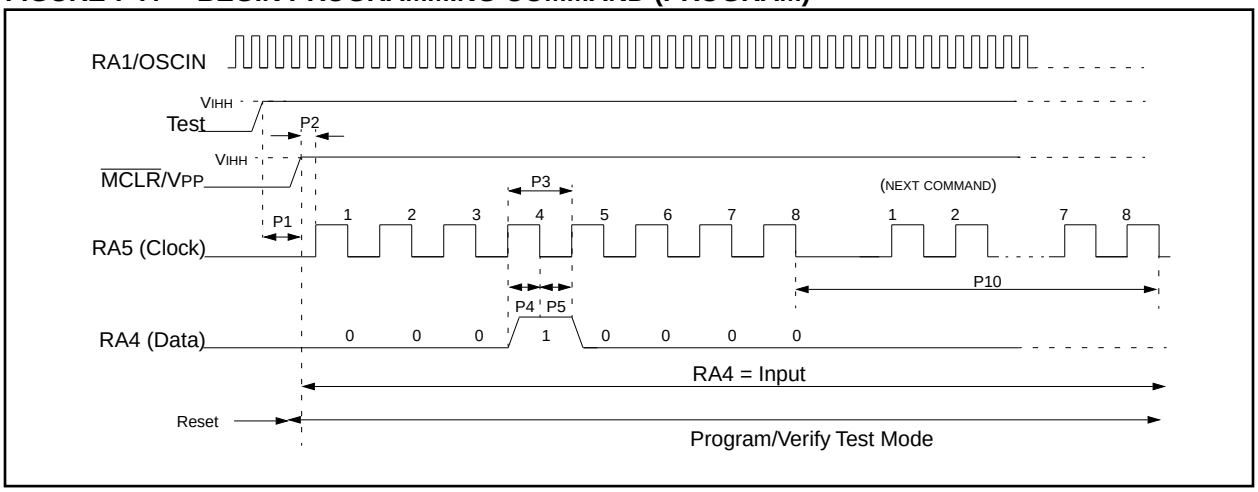


FIGURE 7-7: BEGIN PROGRAMMING COMMAND (PROGRAM)



WORLDWIDE SALES & SERVICE

AMERICAS

Corporate Office
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602-786-7200 Fax: 602-786-7277
Technical Support: 602 786-7627
Web: http://www.microchip.com

Atlanta
Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Boston
Microchip Technology Inc.
5 Mount Royal Avenue
Marlborough, MA 01752
Tel: 508-480-9990 Fax: 508-480-8575

Chicago
Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas
Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 972-991-7177 Fax: 972-991-8588

Dayton
Microchip Technology Inc.
Two Prestige Place, Suite 150
Miamisburg, OH 45342
Tel: 937-291-1654 Fax: 937-291-9175

Los Angeles
Microchip Technology Inc.
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 714-263-1888 Fax: 714-263-1338

New York
Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516-273-5305 Fax: 516-273-5335

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto
Microchip Technology Inc.
5925 Airport Road, Suite 200
Mississauga, Ontario L4V 1W1, Canada
Tel: 905-405-6279 Fax: 905-405-6253

ASIA/PACIFIC

Hong Kong
Microchip Asia Pacific
RM 3801B, Tower Two
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2-401-1200 Fax: 852-2-401-3431

India
Microchip Technology India
No. 6, Legacy, Convent Road
Bangalore 560 025, India
Tel: 91-80-229-0061 Fax: 91-80-229-0062

Korea
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea
Tel: 82-2-554-7200 Fax: 82-2-558-5934

Shanghai
Microchip Technology
RM 406 Shanghai Golden Bridge Bldg.
2077 Yan'an Road West, Hongiao District
Shanghai, PRC 200335
Tel: 86-21-6275-5700
Fax: 86 21-6275-5060

Singapore
Microchip Technology Taiwan
Singapore Branch
200 Middle Road
#10-03 Prime Centre
Singapore 188980
Tel: 65-334-8870 Fax: 65-334-8850

Taiwan, R.O.C
Microchip Technology Taiwan
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2-717-7175 Fax: 886-2-545-0139

EUROPE

United Kingdom
Arizona Microchip Technology Ltd.
Unit 6, The Courtyard
Meadow Bank, Furlong Road
Bourne End, Buckinghamshire SL8 5AJ
Tel: 44-1628-851077 Fax: 44-1628-850259

France
Arizona Microchip Technology SARL
Zone Industrielle de la Bonde
2 Rue du Buisson aux Fraises
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany
Arizona Microchip Technology GmbH
Gustav-Heinemann-Ring 125
D-81739 Müchen, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleone
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-39-6899939 Fax: 39-39-6899883

JAPAN

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shin Yokohama
Kohoku-Ku, Yokohama
Kanagawa 222 Japan
Tel: 81-4-5471- 6166 Fax: 81-4-5471-6122

5/8/97

MICROCHIP

Printed on recycled paper.