The University of Texas at Arlington

Lecture 13
Hardware Connections

CSE 3442/5442

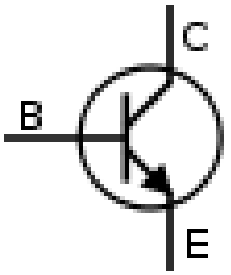
Embedded Systems 1

Based heavily on slides by Dr. Gergely Záruba and Dr. Roger Walker

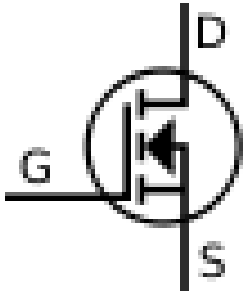
Power Supply Name Conventions

Typical supply pin labeling				
BJT	FET			
V_{CC}	V_{DD}	$V+$	V_{S+}	Positive supply voltage
V_{EE}	V_{SS}	$V-$	V_{S-}	Negative supply voltage

BJT
(N-channel)



MOSFET
(N-channel)

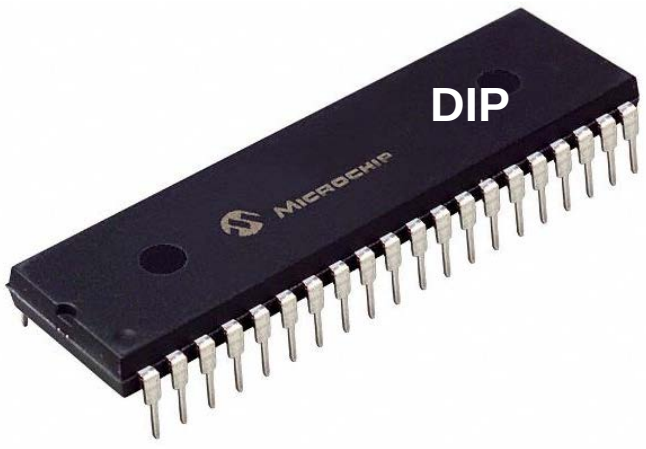


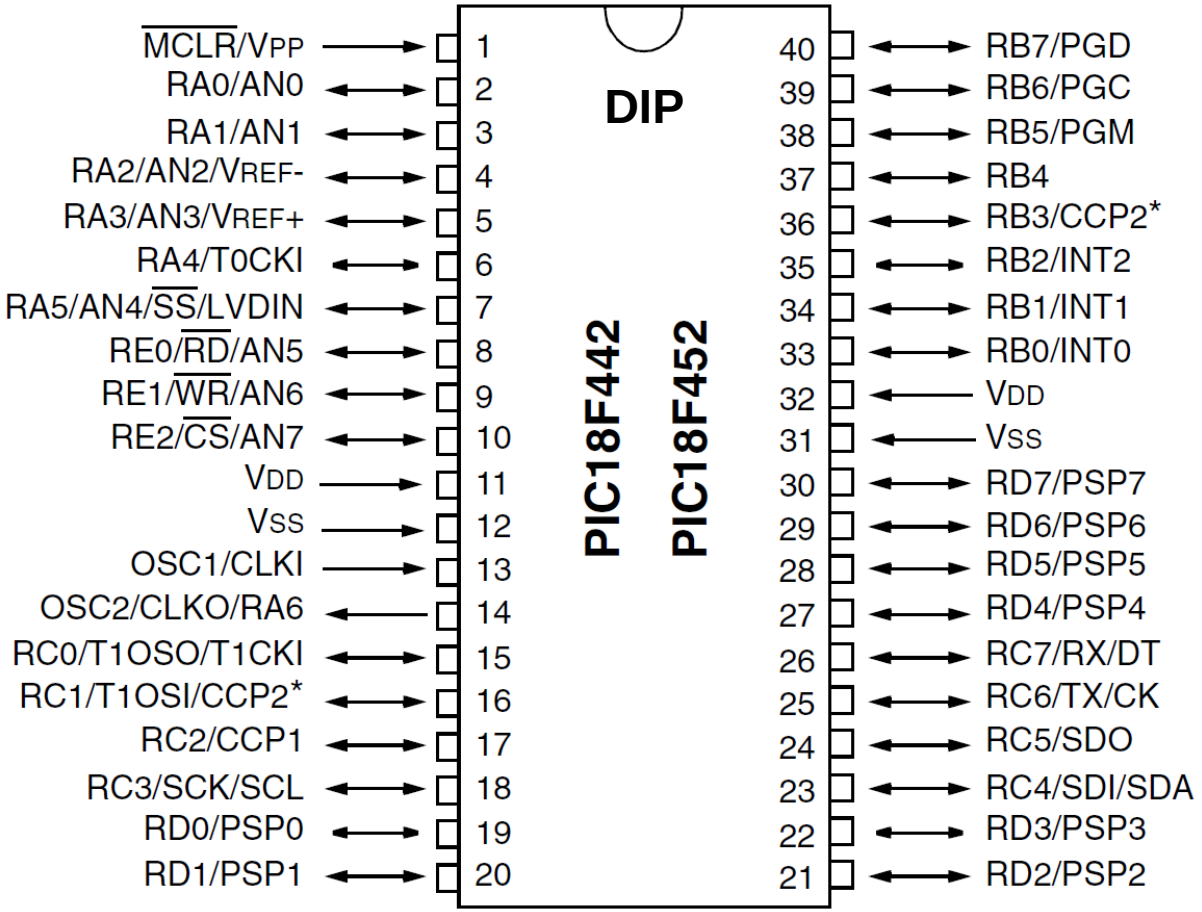
PIC18 Packaging Through-Hole

SIP
Single In-Line Package

DIP
Dual In-Line Package

QIP
Quadruple In-Line Package





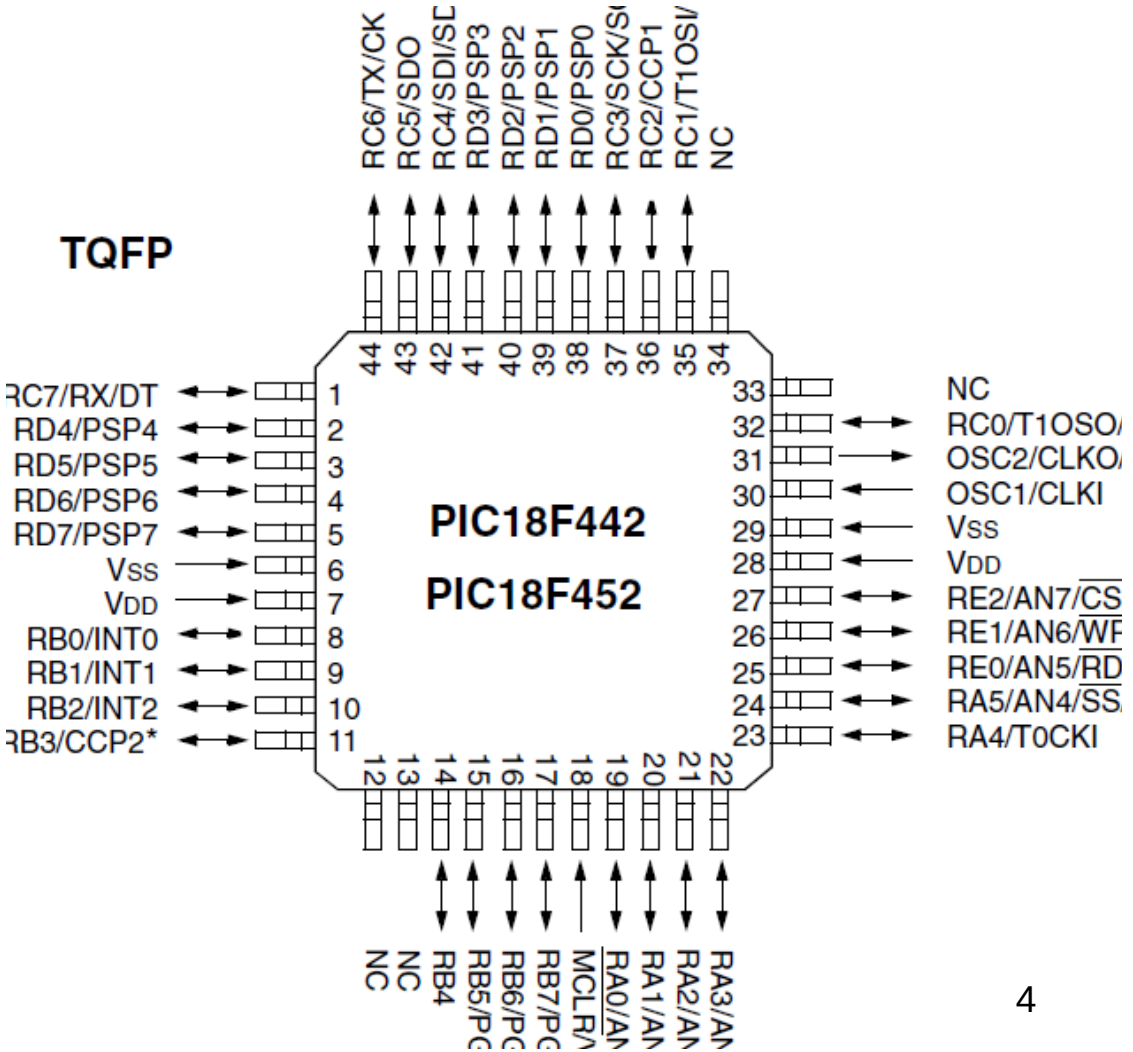
PIC18 Packaging
Surface Mount

SMD
Surface Mount Device

SMT
Surface Mount Technology

PLCC
Plastic Leaded Chip Carrier

TQFP
Thin Quad Flat Pack



Product/Device Naming

PIC18F2420/2520/4420/4520 PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	X	/XX	XXX
Device	Temperature Range	Package	Pattern

Device	PIC18F2420/2520(1), PIC18F4420/4520(1), PIC18F2420/2520T(2), PIC18F4420/4520T(2); VDD range 4.2V to 5.5V PIC18LF2420/2520(1), PIC18LF4420/4520(1), PIC18LF2420/2520T(2), PIC18LF4420/4520T(2); VDD range 2.0V to 5.5V
Temperature Range	I = -40°C to +85°C (Industrial) E = -40°C to +125°C (Extended)
Package	PT = TQFP (Thin Quad Flatpack) SO = SOIC SP = Skinny Plastic DIP P = PDIP ML = QFN
Pattern	QTP, SQTP, Code or Special Requirements (blank otherwise)

Examples:

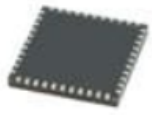
a) PIC18LF4520-I/P 301 = Industrial temp., PDIP package, Extended VDD limits, QTP pattern #301.

b) PIC18LF2420-I/SO = Industrial temp., SOIC package, Extended VDD limits.

c) PIC18F4420-I/P = Industrial temp., PDIP package, normal VDD limits.

Note 1: F = Standard Voltage Range
LF = Wide Voltage Range
2: T = in tape and reel TQFP packages only.

Product/Device Naming

Image	Mouser Part #	Mfr. Part #	Mfr.	Description		Availability	Pricing (USD)
 Enlarge	579-PIC18F4520-I/PT	PIC18F4520-I/PT Available in MultiSIM BLUE	Microchip Technology	8-bit Microcontrollers - MCU 32KB 1536 RAM 36I/O	Datasheet	3,691 In Stock Alternative Packaging	1: $4.89 10: $4.07 25: $3.97 100: $3.88
 Enlarge	579-PIC18F4520-E/PT	PIC18F4520-E/PT Available in MultiSIM BLUE	Microchip Technology	8-bit Microcontrollers - MCU 32KB 1536 RAM 36I/O	Datasheet	2,564 In Stock	1: $5.37 10: $4.47 25: $4.37 100: $4.27
 Enlarge	579-PIC18F4520-I/P	PIC18F4520-I/P Available in MultiSIM BLUE	Microchip Technology	8-bit Microcontrollers - MCU 32KB 1536 RAM 36I/O	Datasheet Product Info.	1,373 In Stock	1: $4.98 10: $4.15 25: $4.05 100: $3.95
 Enlarge	579-PIC18F4520-I/ML	PIC18F4520-I/ML Available in MultiSIM BLUE	Microchip Technology	8-bit Microcontrollers - MCU 32KB 1536 RAM 36I/O	Datasheet	407 In Stock Alternative Packaging	1: $5.14 10: $4.28 25: $4.17 100: $4.08

PIC Max/Min Ratings

22.0 ELECTRICAL CHARACTERISTICS

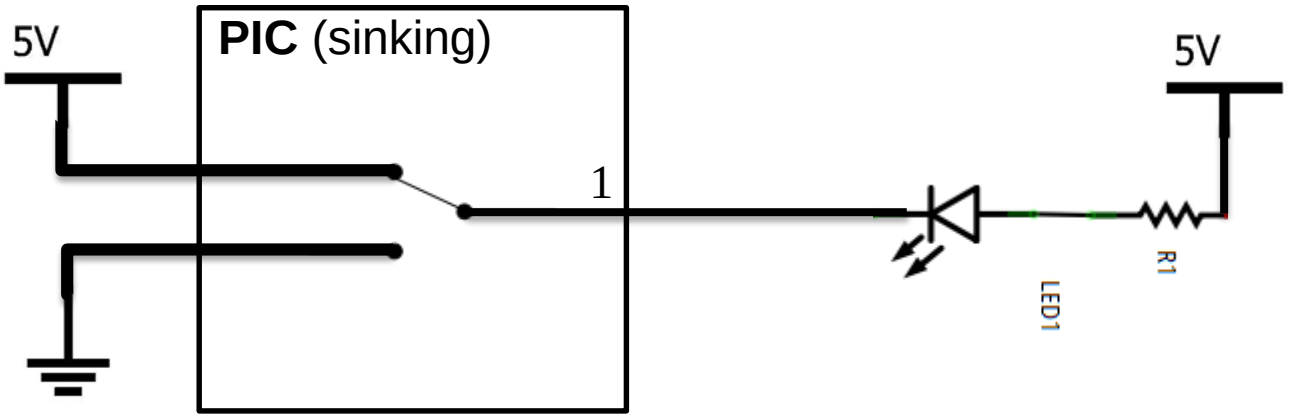
Absolute Maximum Ratings (†)

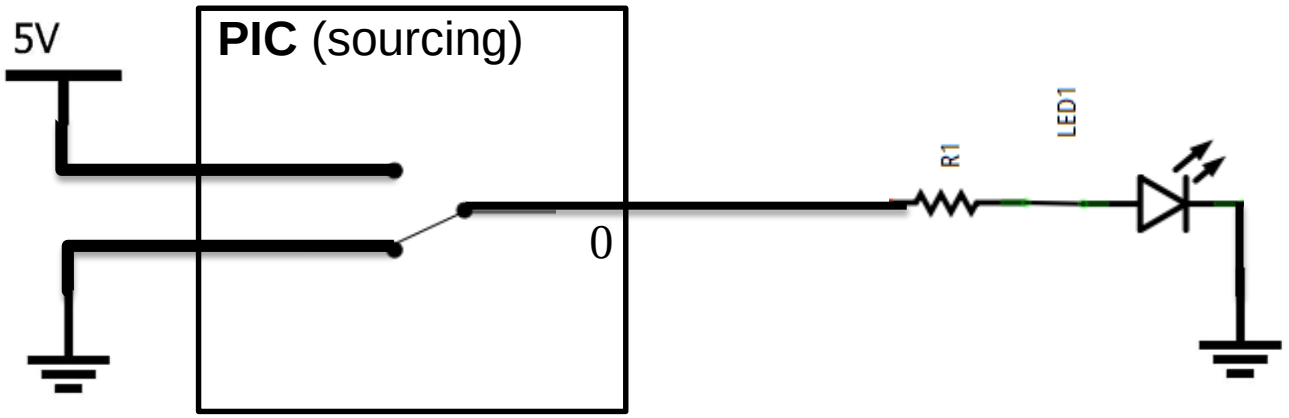
Ambient temperature under bias	-55°C to +125°C
Storage temperature	-65°C to +150°C
Voltage on any pin with respect to Vss (except VDD, $\overline{\text{MCLR}}$, and RA4)	-0.3V to (VDD + 0.3V)
Voltage on VDD with respect to Vss	-0.3V to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to Vss **(Note 2)**	0V to +13.25V
Voltage on RA4 with respect to Vss	0V to +8.5V
Total power dissipation **(Note 1)**	1.0W
Maximum current out of Vss pin	300 mA
Maximum current into VDD pin	250 mA
Input clamp current, IIK (VI < 0 or VI > VDD)	±20 mA
Output clamp current, IOK (VO < 0 or VO > VDD)	±20 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by PORTA, PORTB, and PORTE **(Note 3)** (combined)	200 mA
Maximum current sourced by PORTA, PORTB, and PORTE **(Note 3)** (combined)	200 mA
Maximum current sunk by PORTC and PORTD **(Note 3)** (combined)	200 mA
Maximum current sourced by PORTC and PORTD **(Note 3)** (combined)	200 mA

Note 1: Power dissipation is calculated as follows:
Pdis = VDD x {IDD - ∑ IOH} + ∑ {(VDD-VOH) x IOH} + ∑(Vol x IOL)

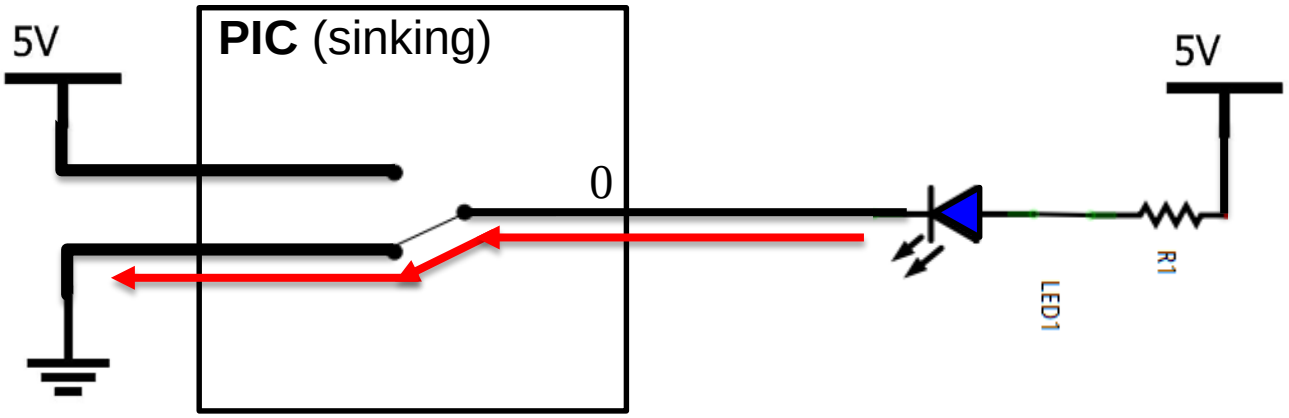
2: Voltage spikes below Vss at the $\overline{\text{MCLR}}$/VPP pin, inducing currents greater than 80 mA, may cause latchup. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{\text{MCLR}}$/VPP pin, rather than pulling this pin directly to Vss.

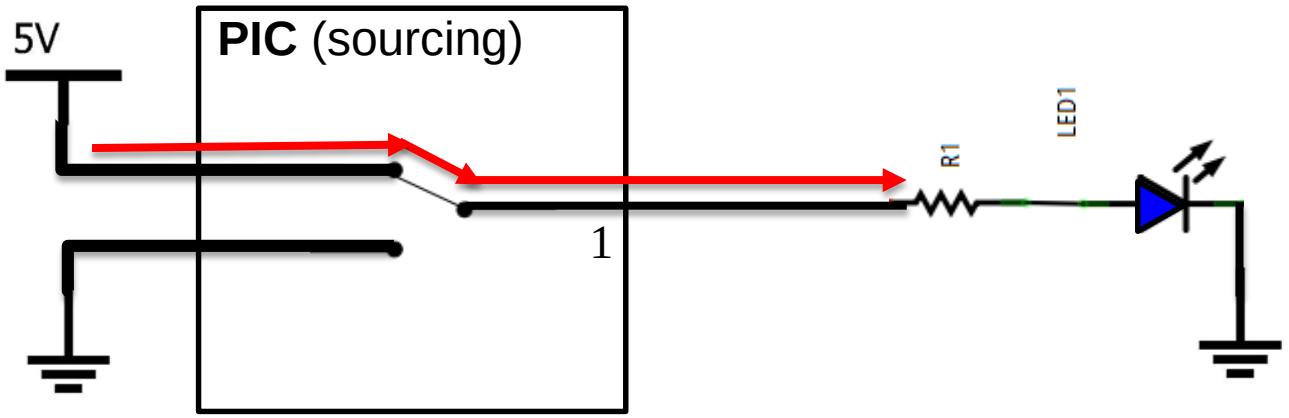
Sink vs. Source



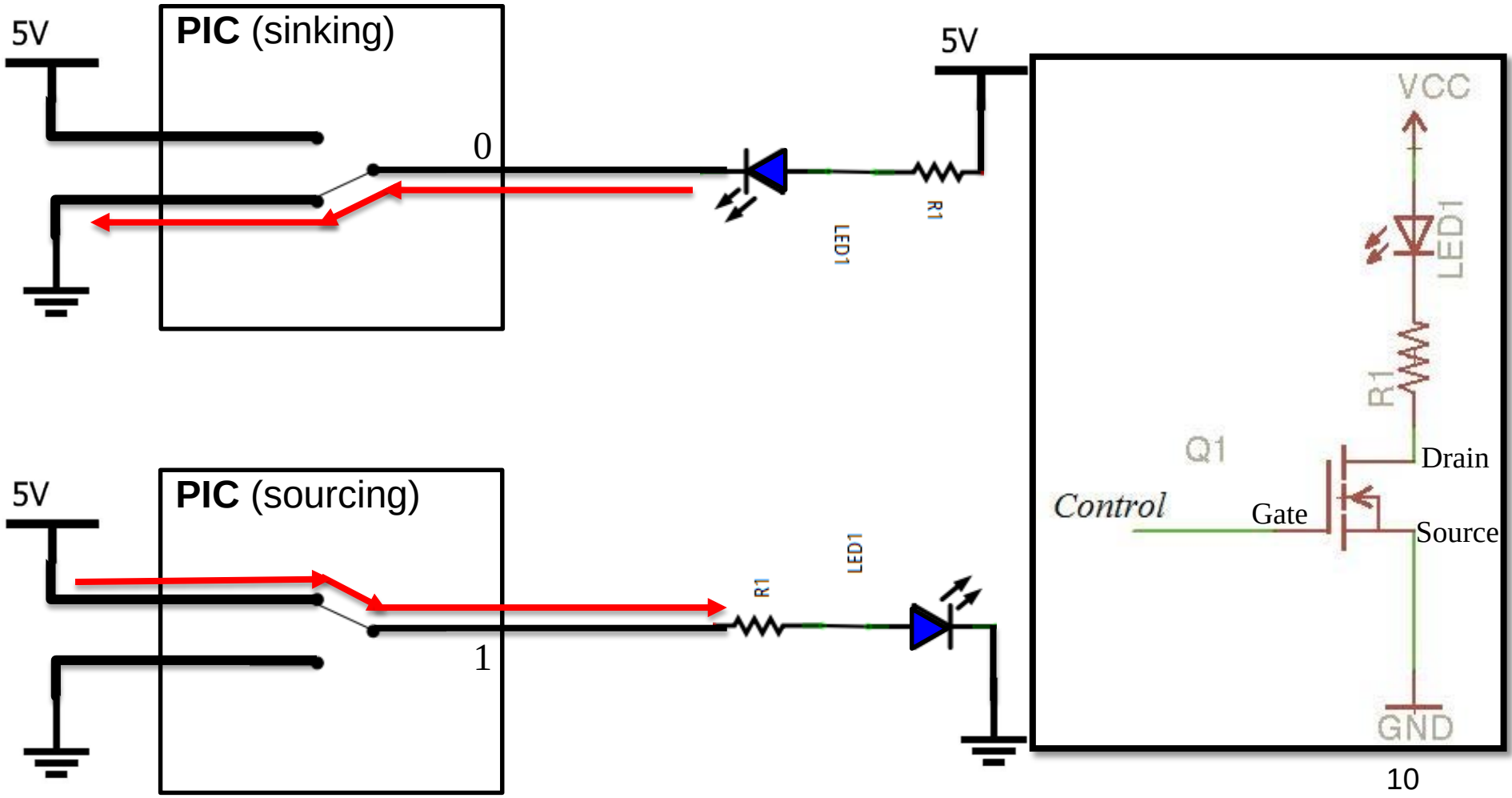


Sink vs. Source





Sink vs. Source



PIC Max/Min Ratings

22.0 ELECTRICAL CHARACTERISTICS

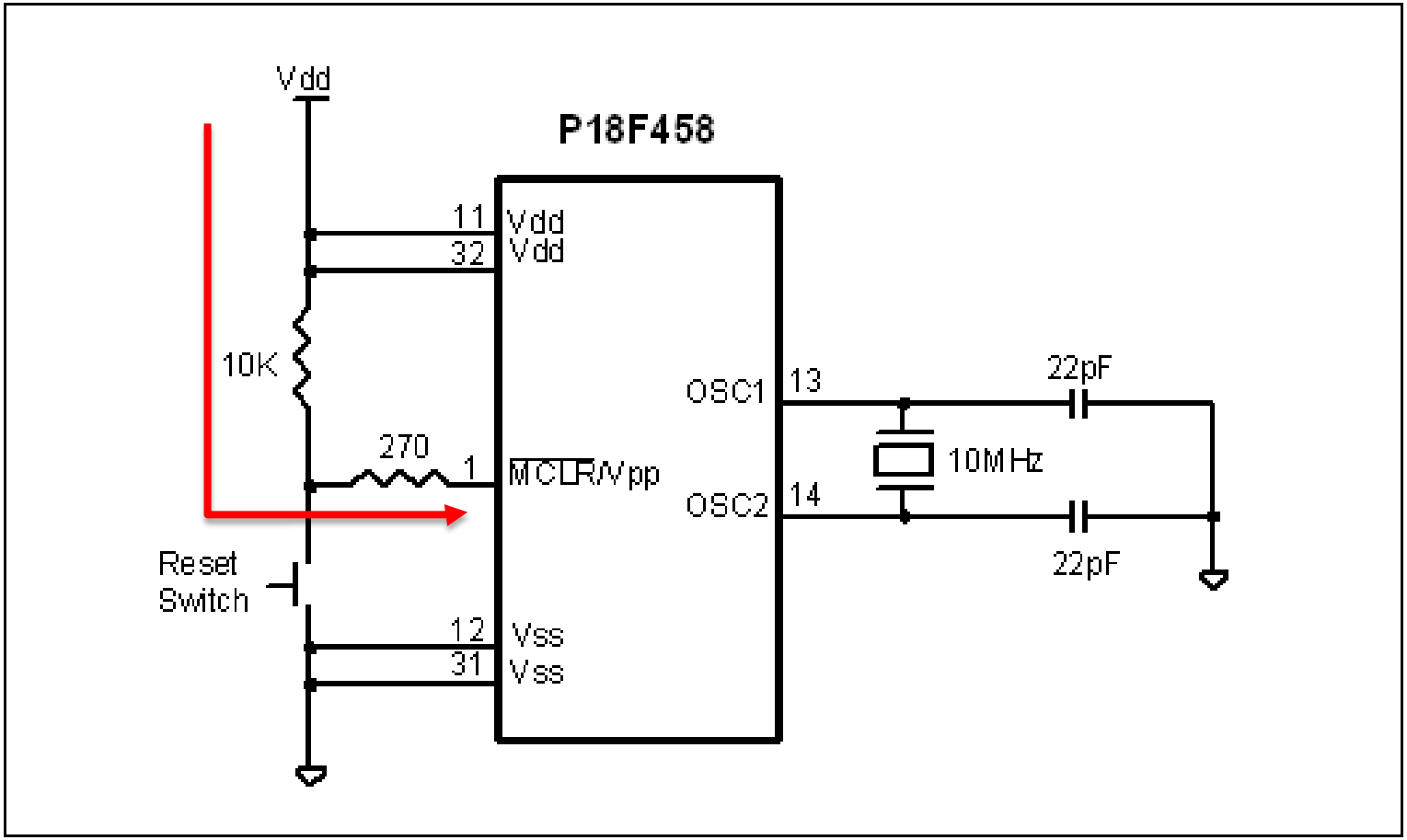
Absolute Maximum Ratings (†)

Ambient temperature under bias	-55°C to +125°C
Storage temperature	-65°C to +150°C
Voltage on any pin with respect to Vss (except VDD, $\overline{\text{MCLR}}$, and RA4)	-0.3V to (VDD + 0.3V)
Voltage on VDD with respect to Vss	-0.3V to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to Vss **(Note 2)**	0V to +13.25V
Voltage on RA4 with respect to Vss	0V to +8.5V
Total power dissipation **(Note 1)**	1.0W
Maximum current out of Vss pin	300 mA
Maximum current into VDD pin	250 mA
Input clamp current, IIK (VI < 0 or VI > VDD)	±20 mA
Output clamp current, IOK (VO < 0 or VO > VDD)	±20 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by PORTA, PORTB, and PORTE **(Note 3)** (combined)	200 mA
Maximum current sourced by PORTA, PORTB, and PORTE **(Note 3)** (combined)	200 mA
Maximum current sunk by PORTC and PORTD **(Note 3)** (combined)	200 mA
Maximum current sourced by PORTC and PORTD **(Note 3)** (combined)	200 mA

Note 1: Power dissipation is calculated as follows:
Pdis = VDD x {IDD - ∑ IOH} + ∑ {(VDD-VOH) x IOH} + ∑(Vol x IOL)

2: Voltage spikes below Vss at the $\overline{\text{MCLR}}$/VPP pin, inducing currents greater than 80 mA, may cause latchup. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{\text{MCLR}}$/VPP pin, rather than pulling this pin directly to Vss.

Powering Up & MCLR PIC18F458



Powering Up & MCLR PIC18F458

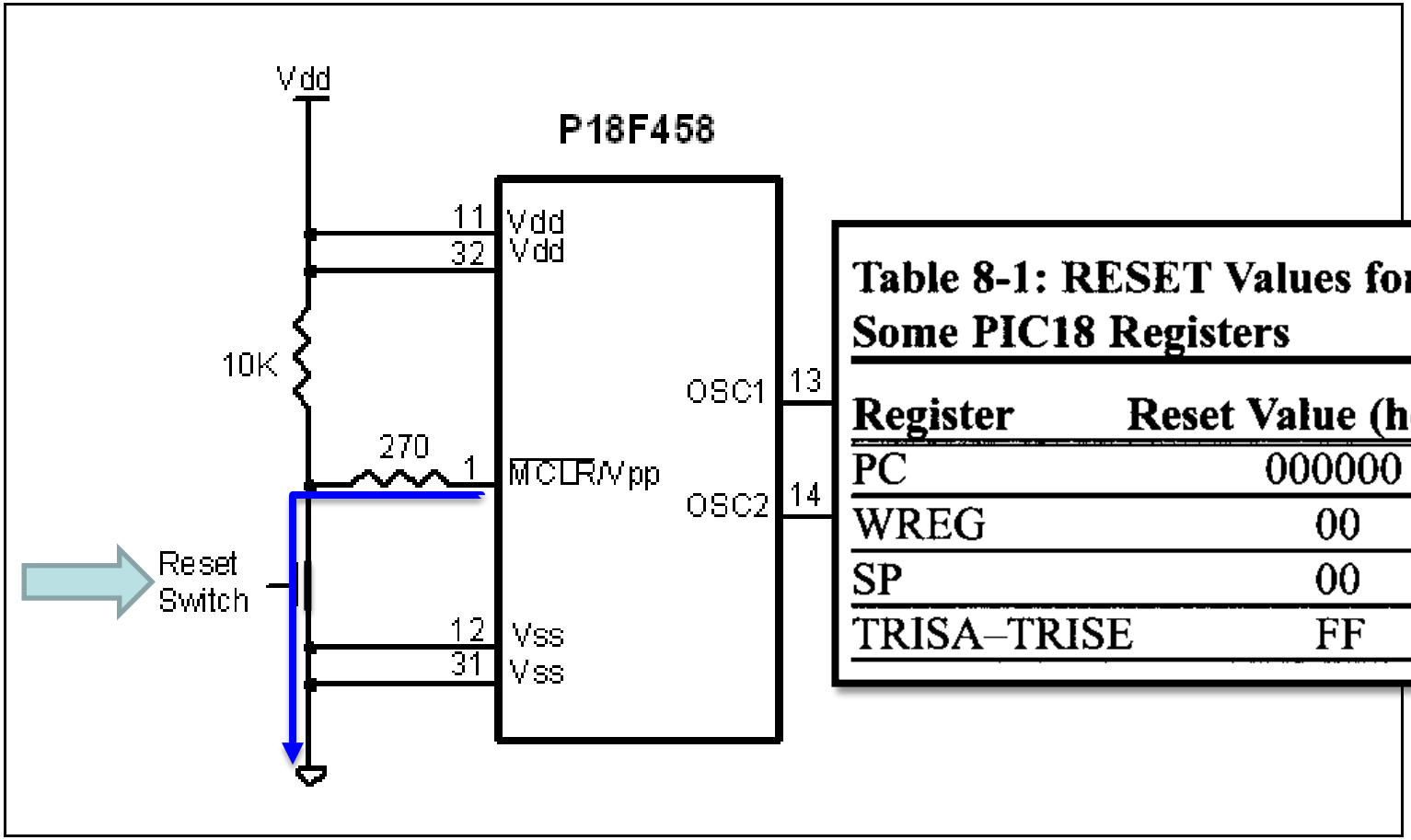


Table 8-1: RESET Values for Some PIC18 Registers

Register	Reset Value (hex)
PC	000000
WREG	00
SP	00
TRISA–TRISE	FF

SFR Values Upon Resets

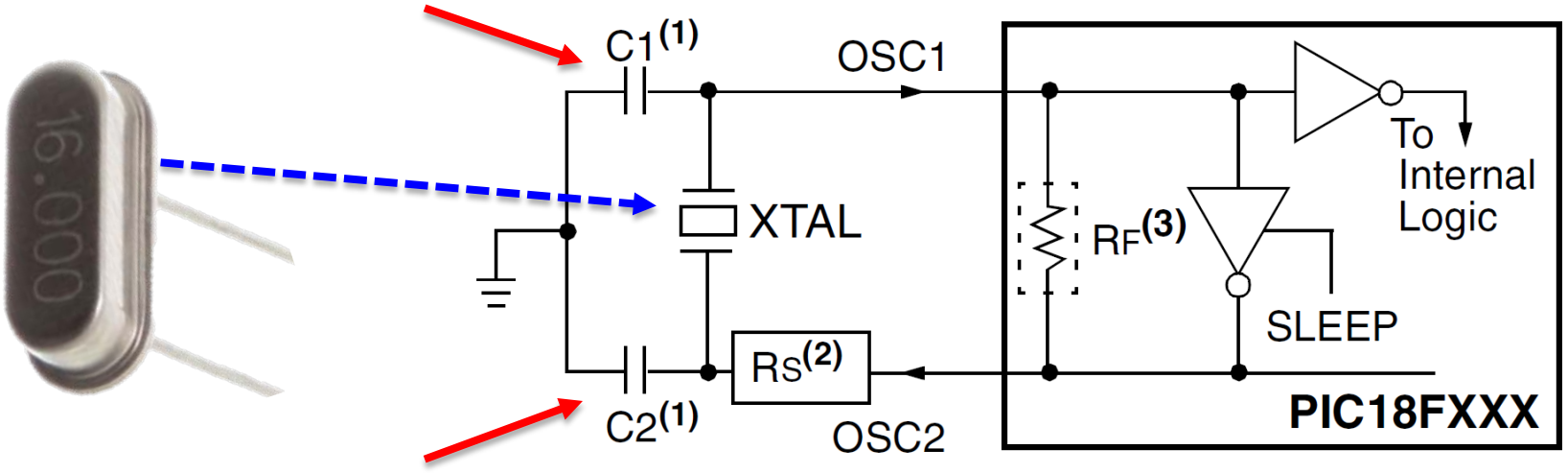
TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices				Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
ADRESH	242	442	252	452	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESL	242	442	252	452	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	242	442	252	452	0000 00-0	0000 00-0	uuuu uu-u
ADCON1	242	442	252	452	00-- 0000	00-- 0000	uu-- uuuu
CCPR1H	242	442	252	452	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	242	442	252	452	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	242	442	252	452	--00 0000	--00 0000	--uu uuuu
CCPR2H	242	442	252	452	xxxx xxxx	uuuu uuuu	uuuu uuuu
...							
EEDATA	242	442	252	452	0000 0000	0000 0000	uuuu uuuu
EECON1	242	442	252	452	xx-0 x000	uu-0 u000	uu-0 u000
EECON2	242	442	252	452	---- ----	---- ----	---- ----

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition.

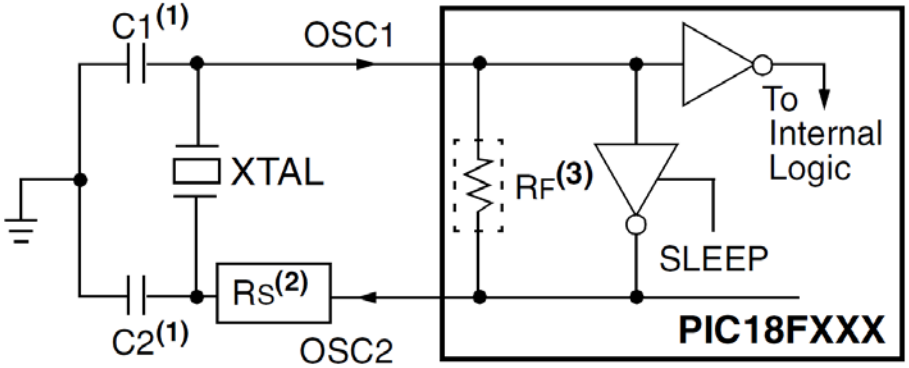
External Crystal

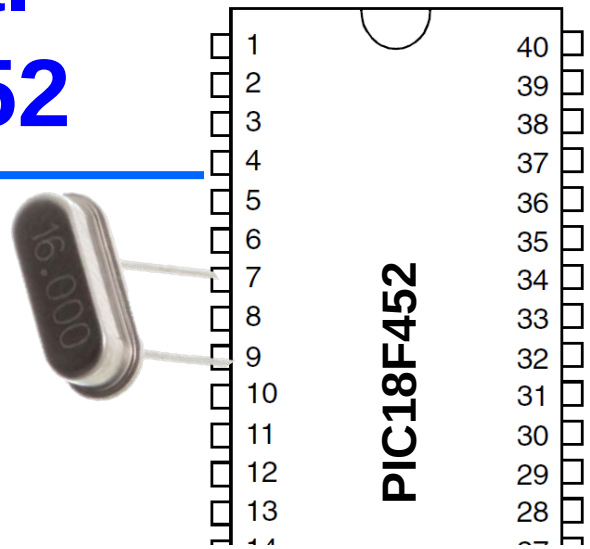
- Reason for parallel capacitors?



- **Great video:** https://www.youtube.com/watch?v=5StwZCeNzVU
- **LC Osc Basics:** https://www.electronics-tutorials.ws/oscillator/oscillators.html
- **Crystal Osc:** https://www.electronics-tutorials.ws/oscillator/crystal.html

External Crystal for the PIC18F452



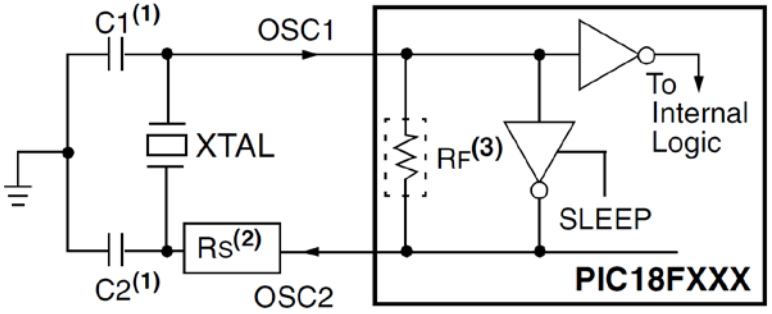


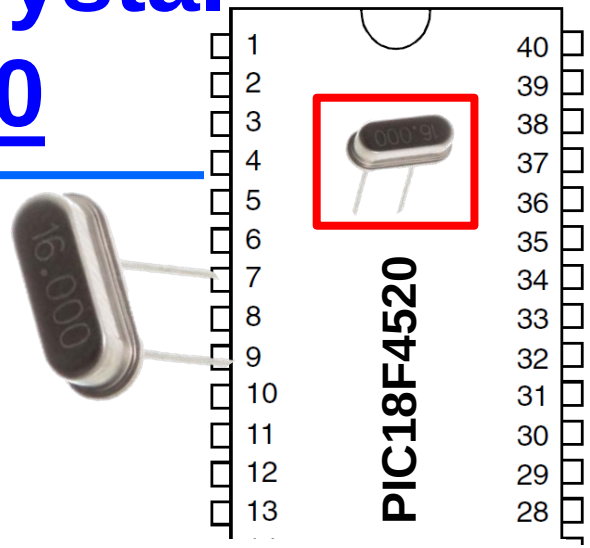
1. LP — Low Power Crystal
2. XT — Crystal/Resonator
3. HS — High Speed Crystal/Resonator
4. HS + PLL — High Speed Crystal/Resonator with PLL enabled
5. RC — External Resistor/Capacitor
6. RCIO — External Resistor/Capacitor with I/O pin enabled
7. EC — External Clock
8. ECIO — External Clock with I/O pin enabled

TABLE 2-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Ranges Tested:			
Mode	**Freq**	**C1**	**C2**
LP	32.0 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	22-68 pF	22-68 pF
	1.0 MHz	15 pF	15 pF
	4.0 MHz	15 pF	15 pF
HS	4.0 MHz	15 pF	15 pF
	8.0 MHz	15-33 pF	15-33 pF
	20.0 MHz	15-33 pF	15-33 pF
	25.0 MHz	15-33 pF	15-33 pF

External or Internal Crystal for the PIC18F4520



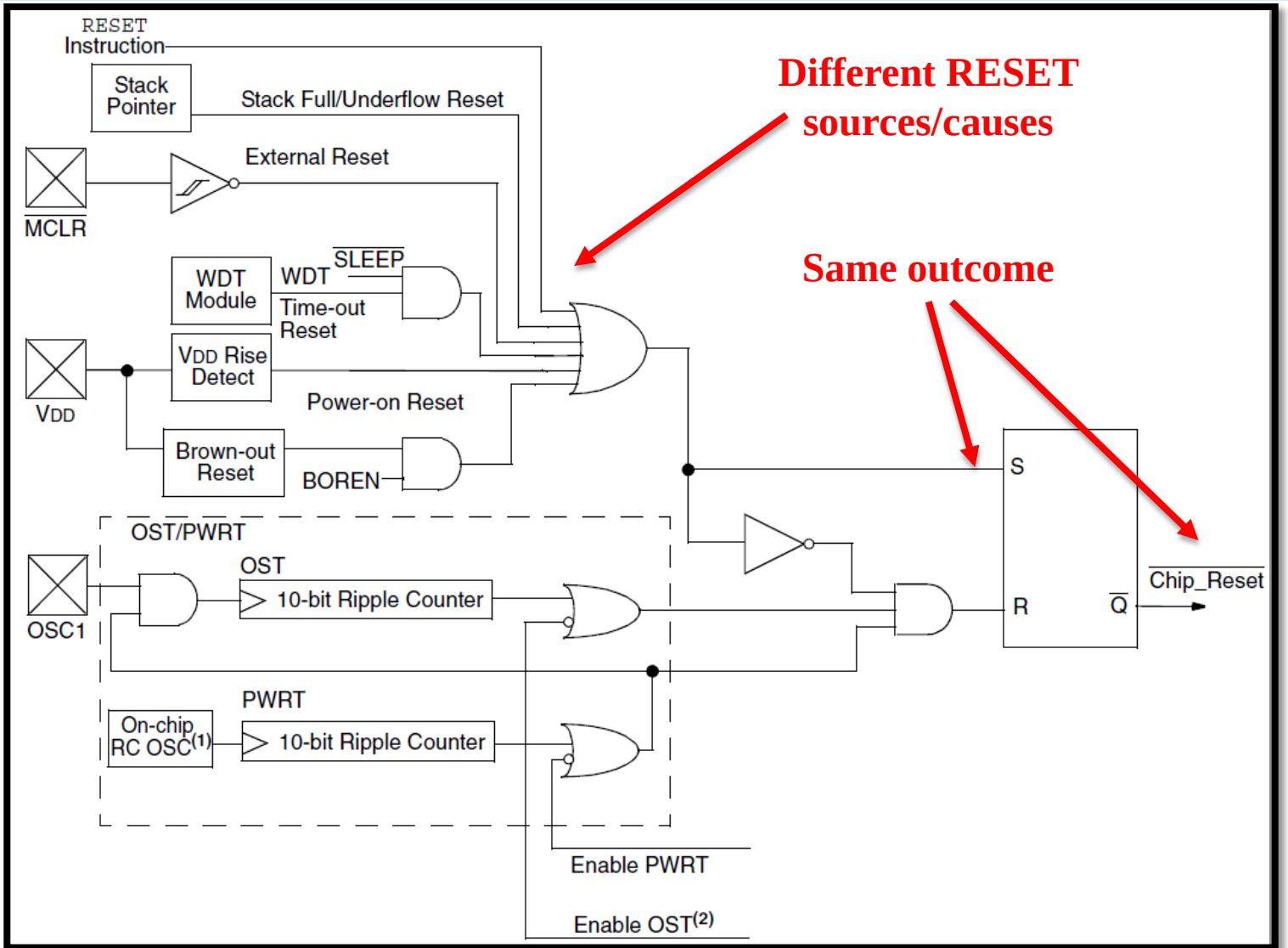


1. LP Low-Power Crystal
2. XT Crystal/Resonator
3. HS High-Speed Crystal/Resonator
4. HSPLL High-Speed Crystal/Resonator with PLL enabled
5. RC External Resistor/Capacitor with Fosc/4 output on RA6
6. RCIO External Resistor/Capacitor with I/O on RA6
7. INTIO1 Internal Oscillator with Fosc/4 output on RA6 and I/O on RA7
8. INTIO2 Internal Oscillator with I/O on RA6 and RA7
9. EC External Clock with Fosc/4 output
10. ECIO External Clock with I/O on RA6

TABLE 2-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

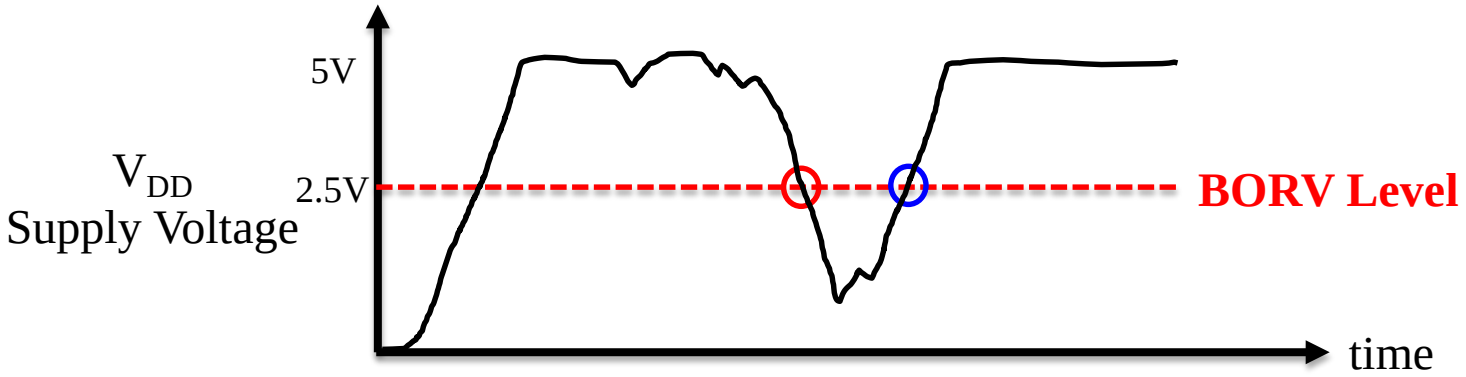
Osc Type	Crystal Freq	Typical Capacitor Values Tested:	
		C1	C2
LP	32 kHz	30 pF	30 pF
XT	1 MHz 4 MHz	15 pF 15 pF	15 pF 15 pF
HS	4 MHz 10 MHz 20 MHz 25 MHz 25 MHz	15 pF 15 pF 15 pF 0 pF 15 pF	15 pF 15 pF 15 pF 5 pF 15 pF

On-Chip Reset Circuit Simplified Diagram (452)

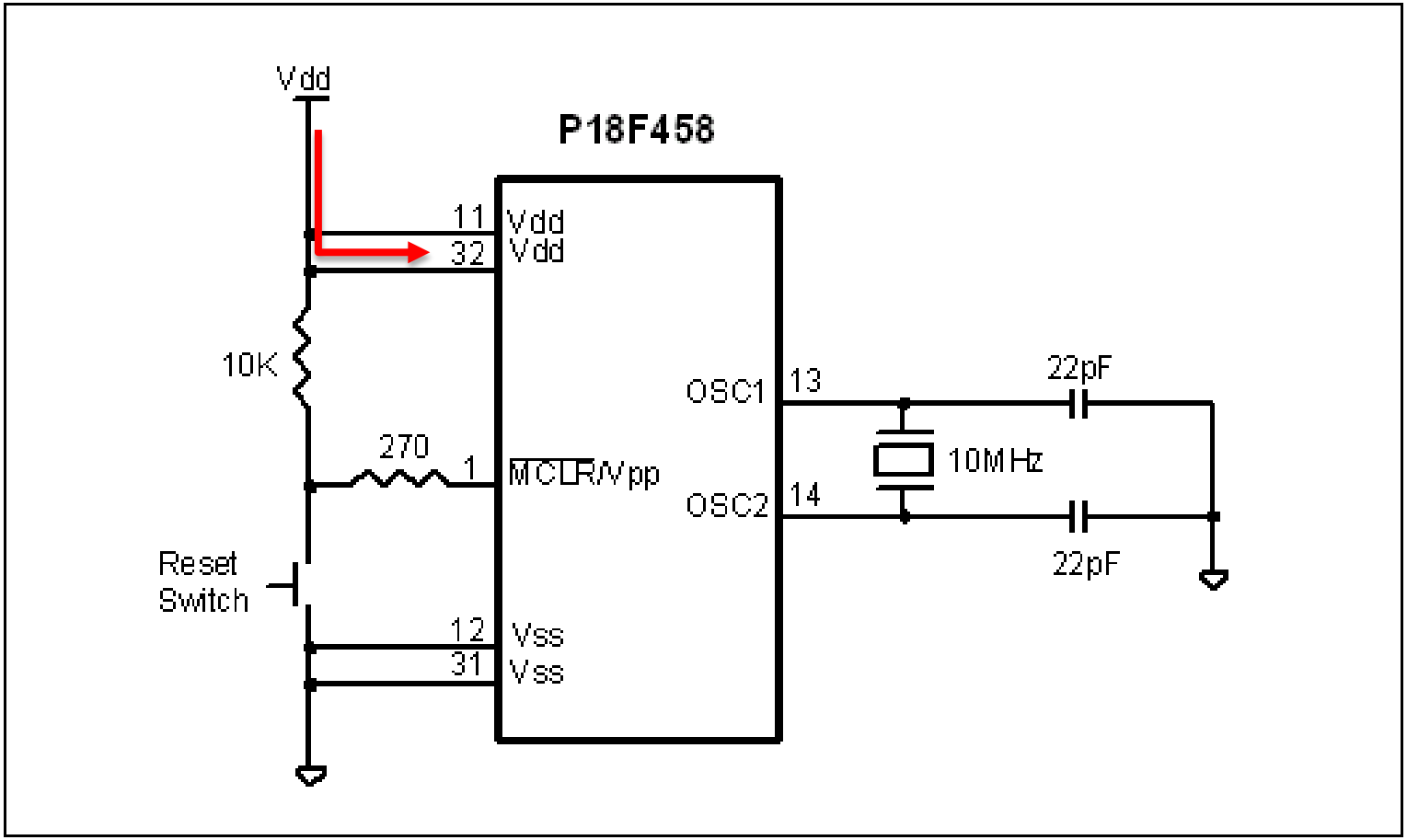


Brown-Out Reset (BOR) Voltage

- Sometimes the V_{DD} (positive supply) can drop below the desired level (+5V)
- This can cause CPU issues and unreliable operation (instr. execution, I/O, ADC, etc.)
- You may specify a **BORV threshold level** where the PIC "resets" automatically if the V_{DD} falls below
- **V_{BOR} options**: 2.5, 2.7, 4.2, and 4.5 V (for 452)



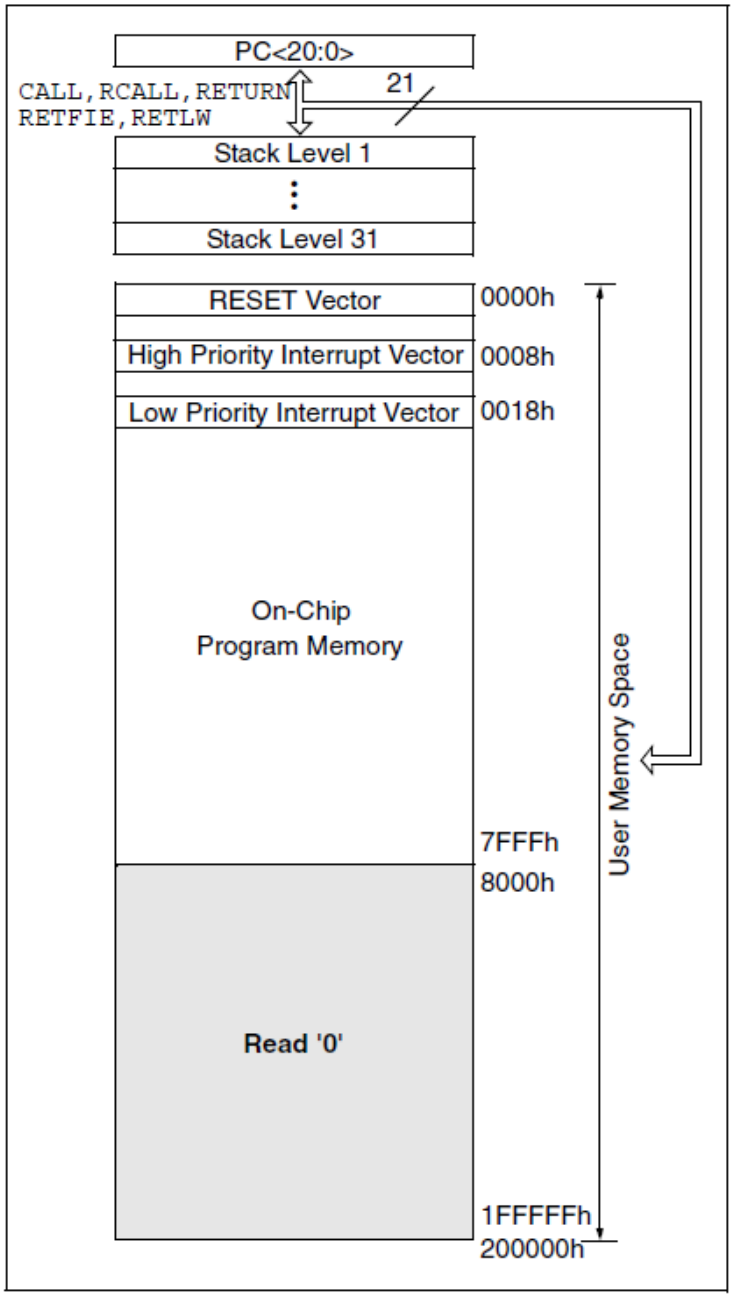
Brown-Out Reset (BOR) Voltage

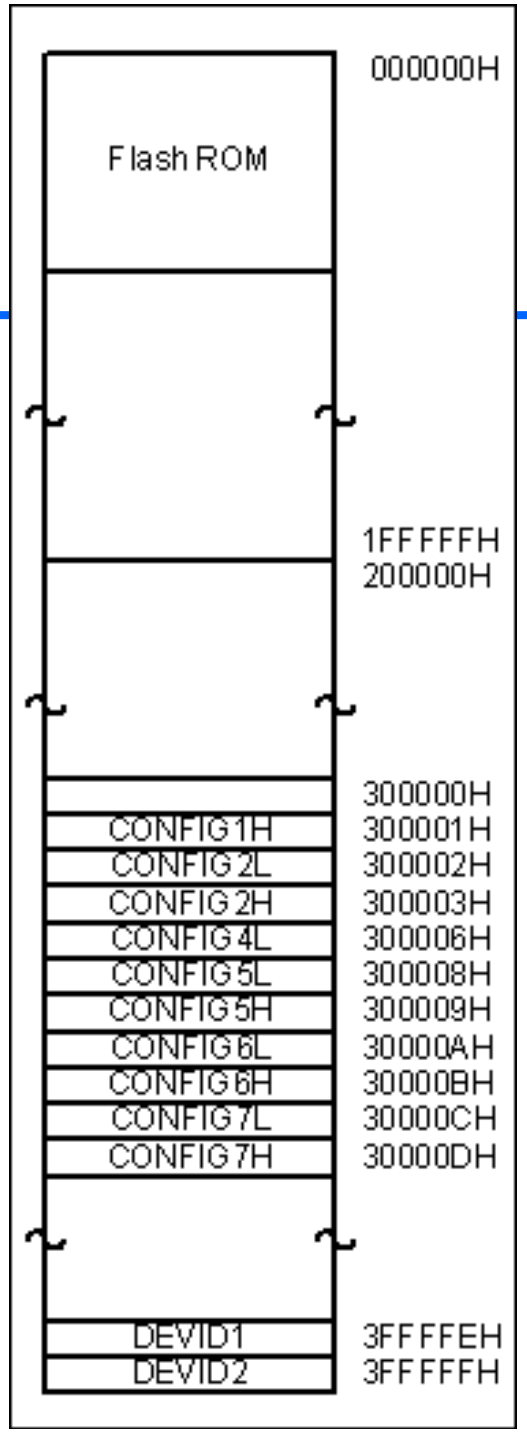


Watchdog Timer (WDT) Reset

- PIC can automatically reset itself when code execution is hung up or non-logical
  - User must clear/restart the WDT periodically
  - Infinite loop, malfunctioning peripherals, etc.
  - Can be used as a makeshift debugger
- Programmer can enable or disable and set the **time-out period**
  - Possible *ms – 100xsec* range
- Clear the WDT (prevent a reset) by using
  - ***CLRWDT*** instruction

FIGURE 4-2: PROGRAM MEMORY MAP AND STACK FOR PIC18F452/252





Configuration Registers 452

TABLE 19-1: CONFIGURATION BITS AND DEVICE IDS

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
300001h	CONFIG1H	—	—	$\overline{\text{OSCSEN}}$	—	—	FOSC2	FOSC1	FOSC0	--1- -111
300002h	CONFIG2L	—	—	—	—	BORV1	BORV0	BOREN	$\overline{\text{PWRTEN}}$	---- 1111
300003h	CONFIG2H	—	—	—	—	WDTPS2	WDTPS1	WDTPS0	WDTEN	---- 1111
300005h	CONFIG3H	—	—	—	—	—	—	—	CCP2MX	---- ---1
300006h	CONFIG4L	$\overline{\text{DEBUG}}$	—	—	—	—	LVP	—	STVREN	1--- -1-1
300008h	CONFIG5L	—	—	—	—	CP3	CP2	CP1	CP0	---- 1111
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—	11-- ----
30000Ah	CONFIG6L	—	—	—	—	WRT3	WRT2	WRT1	WRT0	---- 1111
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	—	—	—	—	—	111- ----
30000Ch	CONFIG7L	—	—	—	—	EBTR3	EBTR2	EBTR1	EBTR0	---- 1111
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—	-1-- ----
3FFFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	(1)
3FFFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	0000 0100

Configuration Registers 4520

TABLE 23-1: CONFIGURATION BITS AND DEVICE IDs

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
300001h	CONFIG1H	IESO	FCMEN	—	—	FOSC3	FOSC2	FOSC1	FOSC0	00-- 0111
300002h	CONFIG2L	—	—	—	BORV1	BORV0	BOREN1	BOREN0	$\overline{\text{PWRTEN}}$	---1 1111
300003h	CONFIG2H	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN	---1 1111
300005h	CONFIG3H	MCLRE	—	—	—	—	LPT1OSC	PBADEN	CCP2MX	1--- -011
300006h	CONFIG4L	$\overline{\text{DEBUG}}$	XINST	—	—	—	LVP	—	STVREN	10-- -1-1
300008h	CONFIG5L	—	—	—	—	CP3(1)	CP2(1)	CP1	CP0	---- 1111
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—	11-- ----
30000Ah	CONFIG6L	—	—	—	—	WRT3(1)	WRT2(1)	WRT1	WRT0	---- 1111
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	—	—	—	—	—	111- ----
30000Ch	CONFIG7L	—	—	—	—	EBTR3(1)	EBTR2(1)	EBTR1	EBTR0	---- 1111
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—	-1-- ----
3FFFFEh	DEVID1(1)	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxxx xxxx(2)
3FFFFFh	DEVID2(1)	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	0000 1100

CONFIG1H - Oscillator

REGISTER 19-1: CONFIGURATION REGISTER 1 HIGH (CONFIG1H: BYTE ADDRESS 300001h)

U-0	U-0	R/P-1	U-0	U-0	R/P-1	R/P-1	R/P-1
—	—	$\overline{\text{OSCSEN}}$	—	—	FOSC2	FOSC1	FOSC0
bit 7							bit 0

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **$\overline{\text{OSCSEN}}$**: Oscillator System Clock Switch Enable bit

1 = Oscillator system clock switch option is disabled (main oscillator is source)
0 = Oscillator system clock switch option is enabled (oscillator switching is enabled)

bit 4-3 **Unimplemented:** Read as '0'

bit 2-0 **FOSC2:FOSC0**: Oscillator Selection bits

111 = RC oscillator w/ OSC2 configured as RA6
110 = HS oscillator with PLL enabled/Clock frequency = (4 x Fosc)
101 = EC oscillator w/ OSC2 configured as RA6
100 = EC oscillator w/ OSC2 configured as divide-by-4 clock output
011 = RC oscillator
010 = HS oscillator
001 = XT oscillator
000 = LP oscillator

CONFIG2L – Initial Transients

REGISTER 19-2: CONFIGURATION REGISTER 2 LOW (CONFIG2L: BYTE ADDRESS 300002h)

U-0	U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1
—	—	—	—	BORV1	BORV0	BOREN	$\overline{\text{PWRTEN}}$
bit 7							bit 0

bit 7-4 **Unimplemented:** Read as '0'

bit 3-2 **BORV1:BORV0:** Brown-out Reset Voltage bits

11 = V_{BOR} set to 2.5V
10 = V_{BOR} set to 2.7V
01 = V_{BOR} set to 4.2V
00 = V_{BOR} set to 4.5V

bit 1 **BOREN:** Brown-out Reset Enable bit

1 = Brown-out Reset enabled
0 = Brown-out Reset disabled

bit 0 **$\overline{\text{PWRTEN}}$:** Power-up Timer Enable bit

1 = PWRT disabled
0 = PWRT enabled

CONFIG2H – Rottweilers

REGISTER 19-3: CONFIGURATION REGISTER 2 HIGH (CONFIG2H: BYTE ADDRESS 300003h)

U-0	U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1
—	—	—	—	WDTPS2	WDTPS1	WDTPS0	WDTEN
bit 7							bit 0

bit 7-4 **Unimplemented:** Read as '0'

bit 3-1 **WDTPS2:WDTPS0:** Watchdog Timer Postscale Select bits

111 = 1:128
110 = 1:64
101 = 1:32
100 = 1:16
011 = 1:8
010 = 1:4
001 = 1:2
000 = 1:1

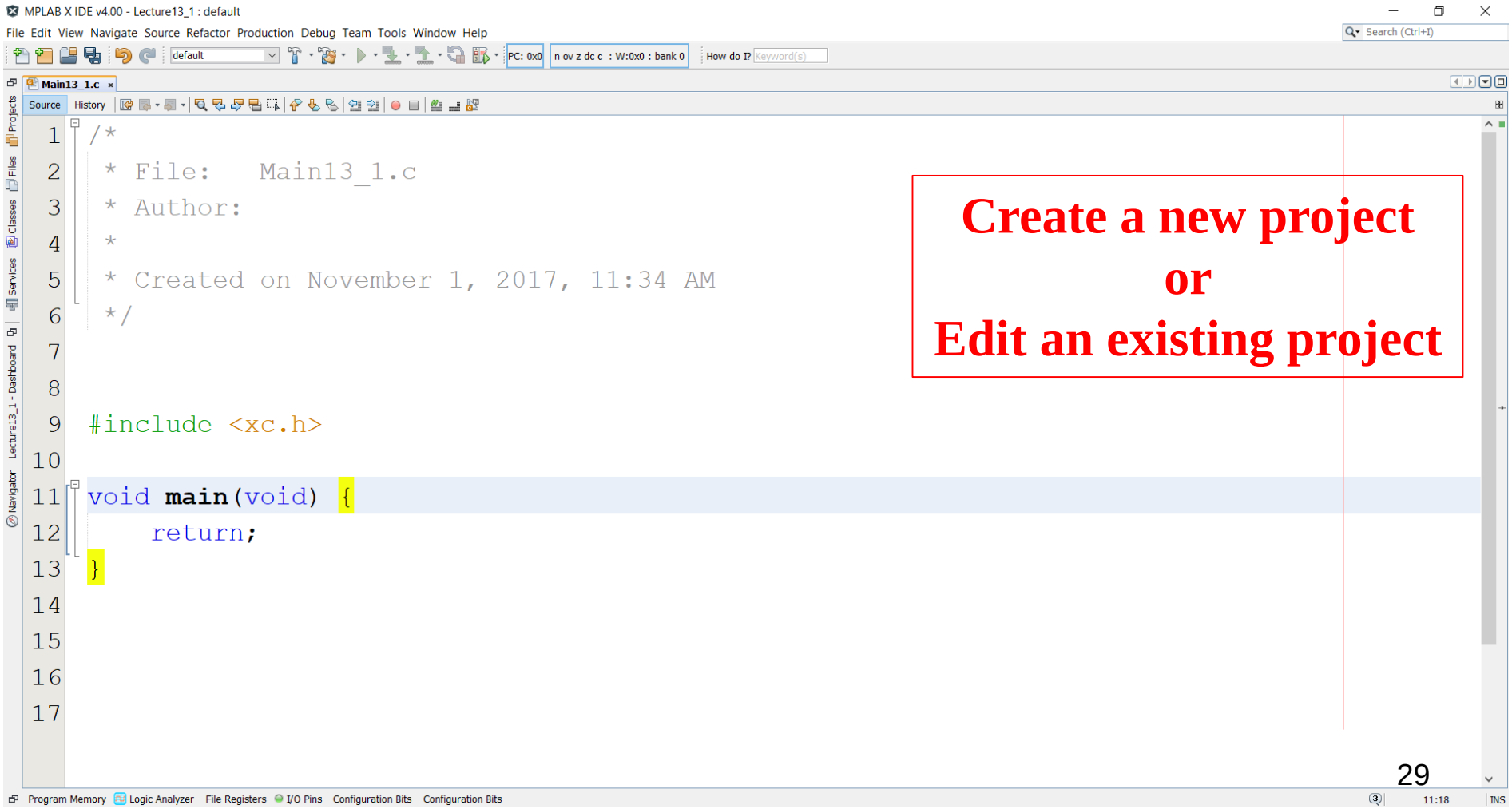
bit 0 **WDTEN:** Watchdog Timer Enable bit

1 = WDT enabled
0 = WDT disabled (control is placed on the SWDTEN bit)

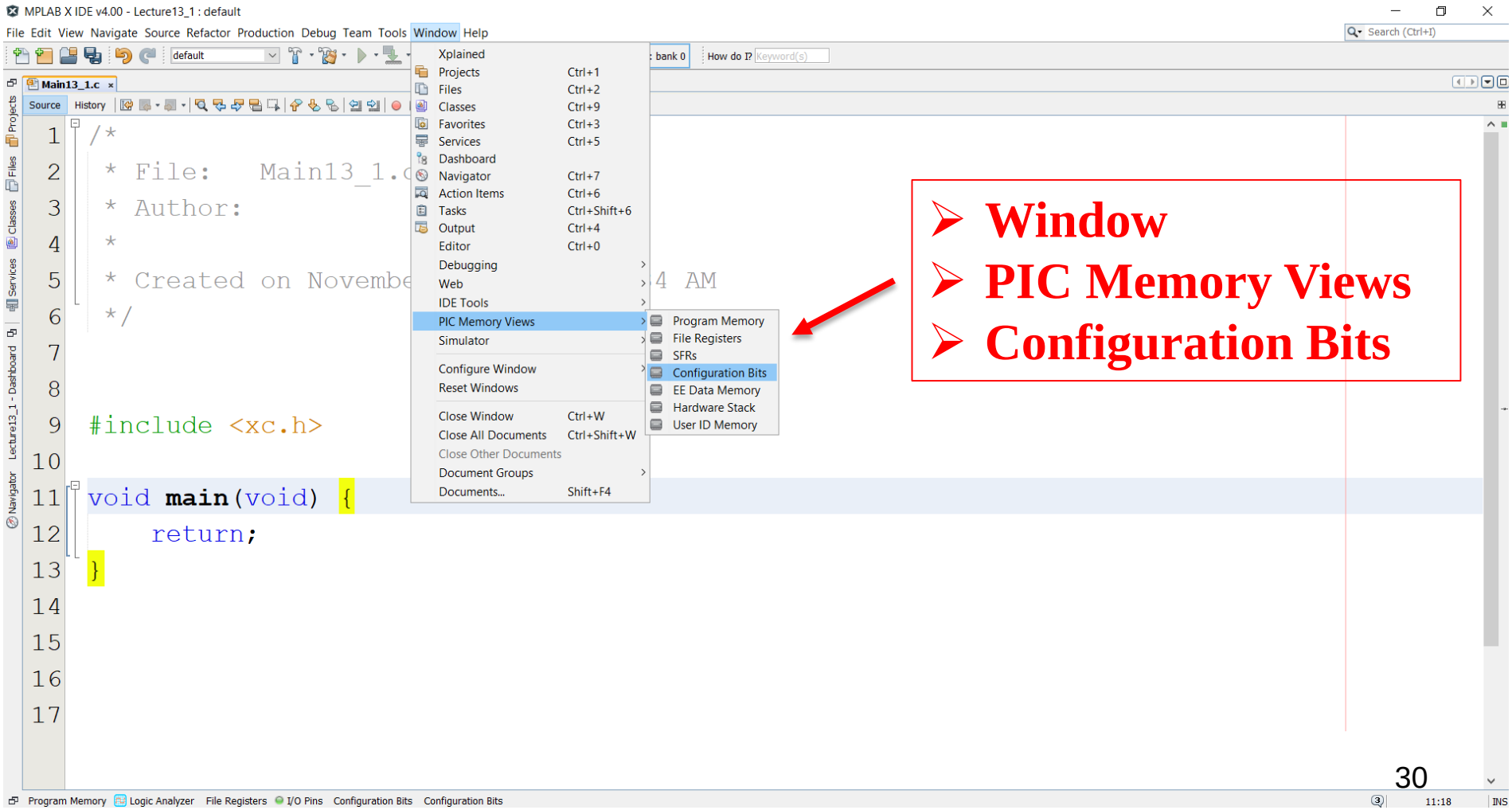
Configuration Bit Settings

```
#pragma config OSC = HS          // Oscillator Selection bits (HS oscillator)
#pragma config FCMEN = OFF       // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monit
#pragma config IESO = OFF        // Internal/External Oscillator Switchover bit (Oscillator S
#pragma config PWRT = OFF        // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOREN = OFF       // Brown-out Reset Enable bits (Brown-out Reset disabled in
#pragma config BORV = 3          // Brown Out Reset Voltage bits (Minimum setting)
#pragma config WDT = OFF         // Watchdog Timer Enable bit (WDT disabled (control is place
#pragma config WDTPS = 32768     // Watchdog Timer Postscale Select bits (1:32768)
#pragma config CCP2MX = PORTC    // CCP2 MUX bit (CCP2 input/output is multiplexed with RC1)
#pragma config PBADEN = OFF      // PORTB A/D Enable bit (PORTB<4:0> pins are configured as d
#pragma config LPT1OSC = OFF     // Low-Power Timer1 Oscillator Enable bit (Timer1 configured
#pragma config MCLRE = ON        // MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin disa
```

Config Bit Settings
MPLAB X IDE

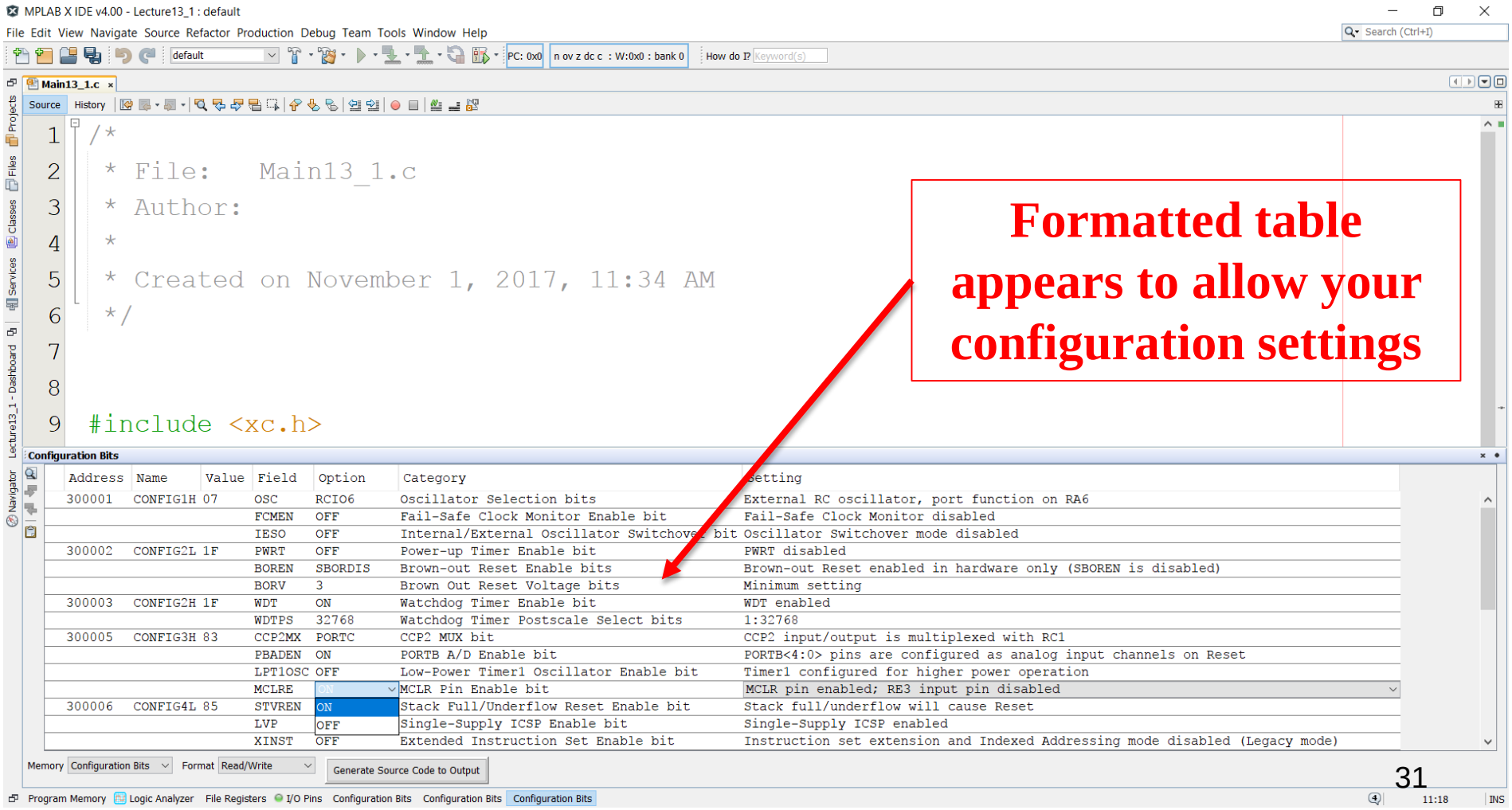


Config Bit Settings
MPLAB X IDE



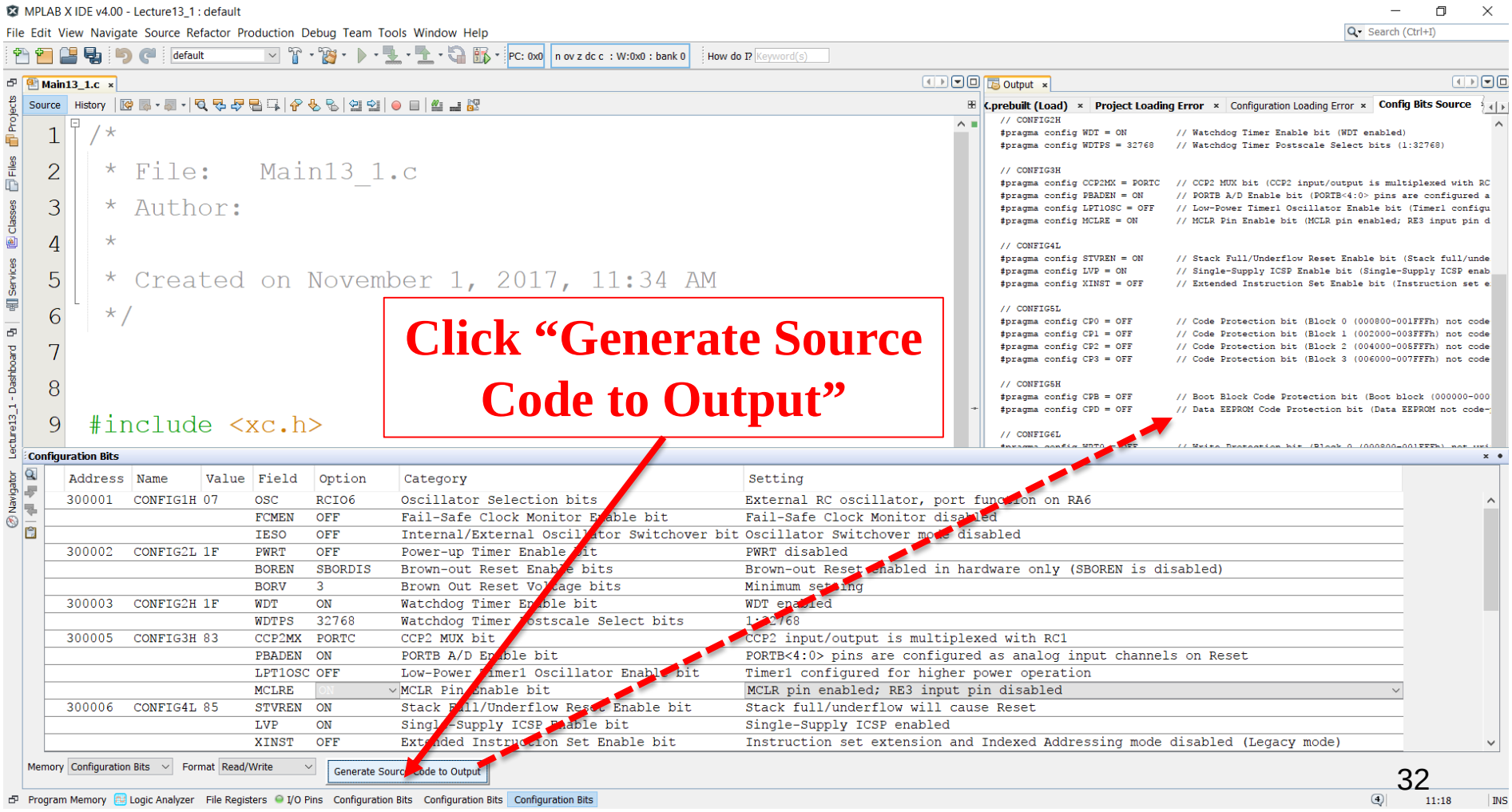
- Window
- PIC Memory Views
- Configuration Bits

Config Bit Settings
MPLAB X IDE

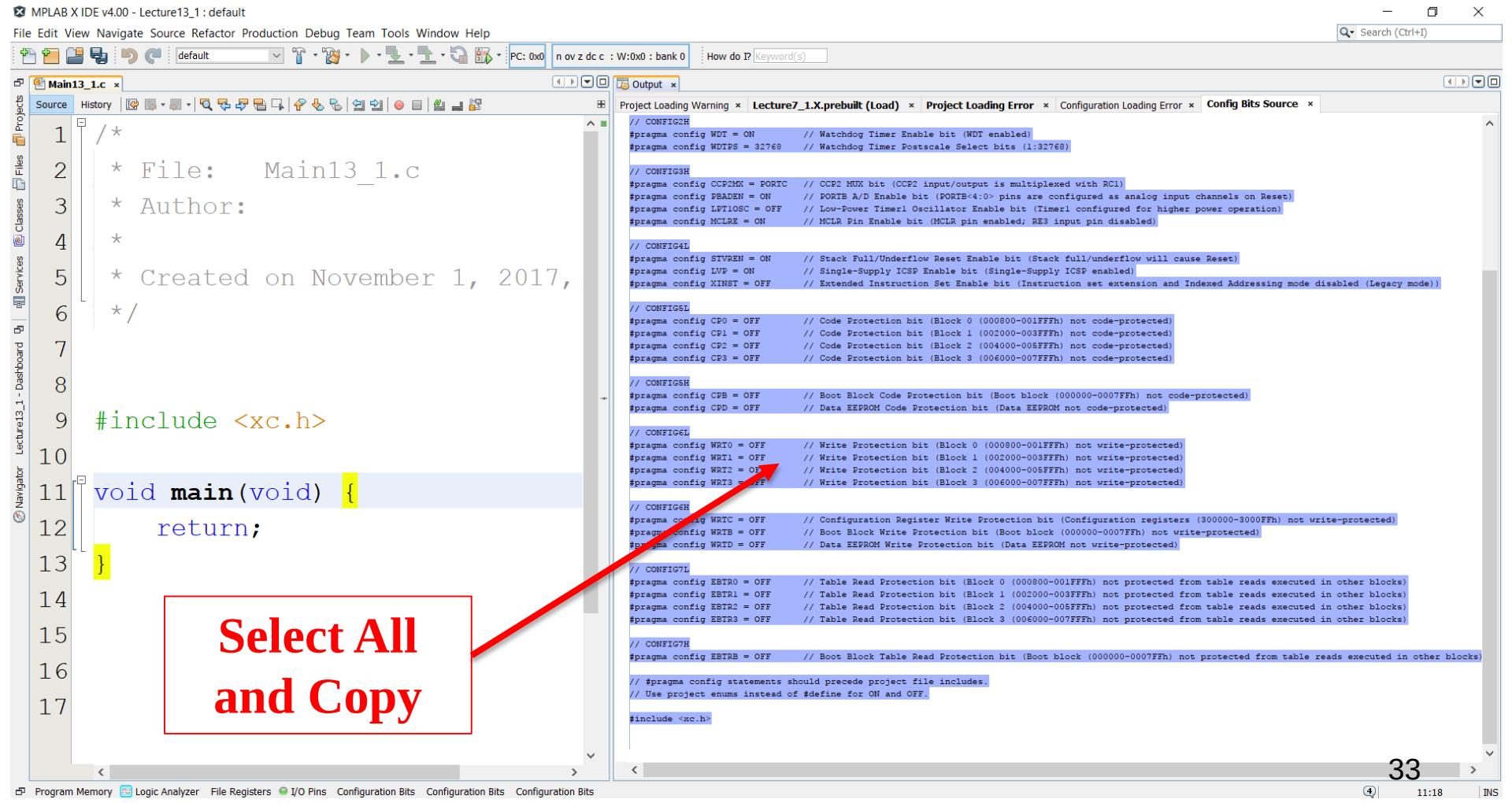


Formatted table appears to allow your configuration settings

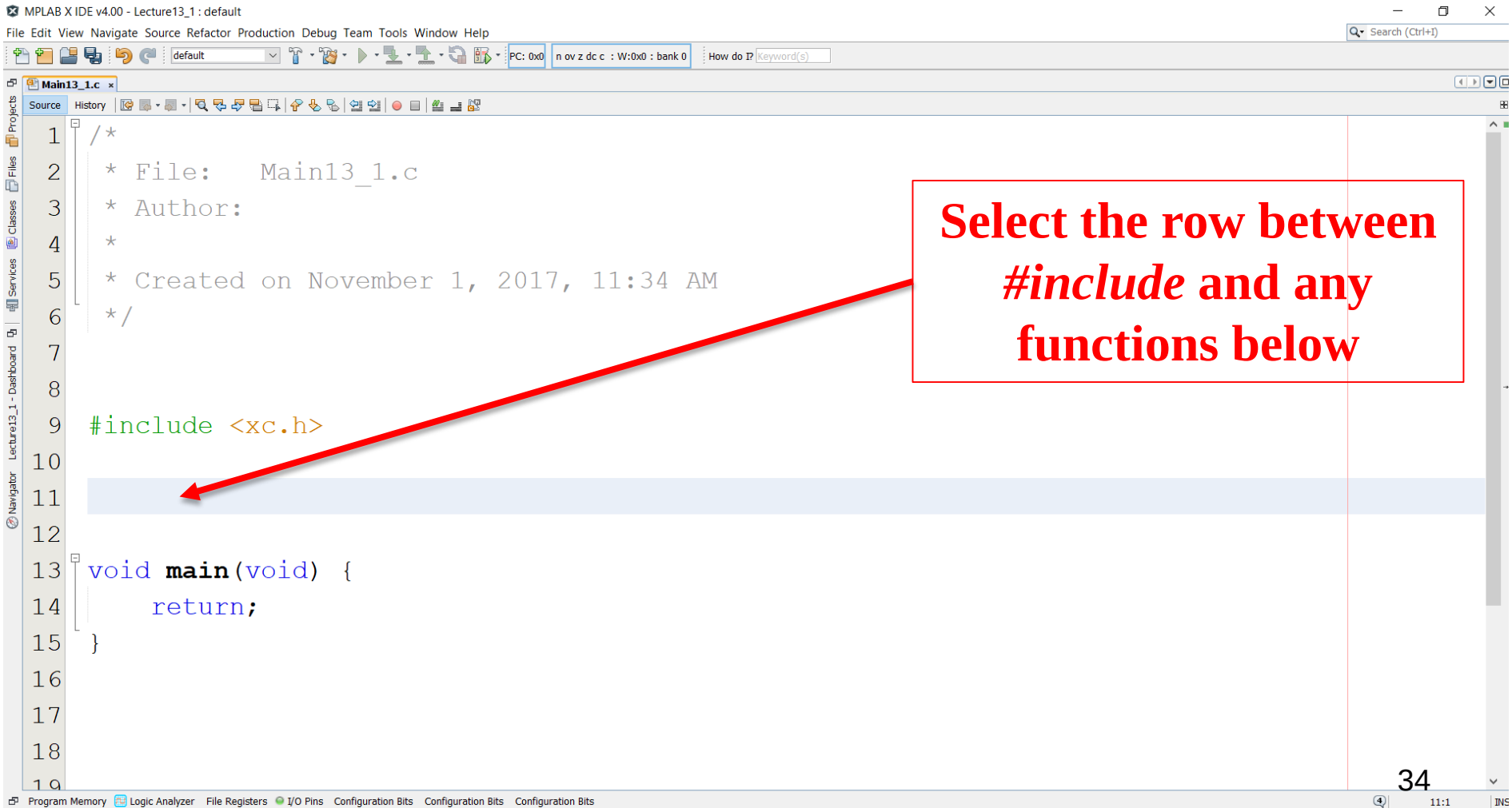
Config Bit Settings
MPLAB X IDE



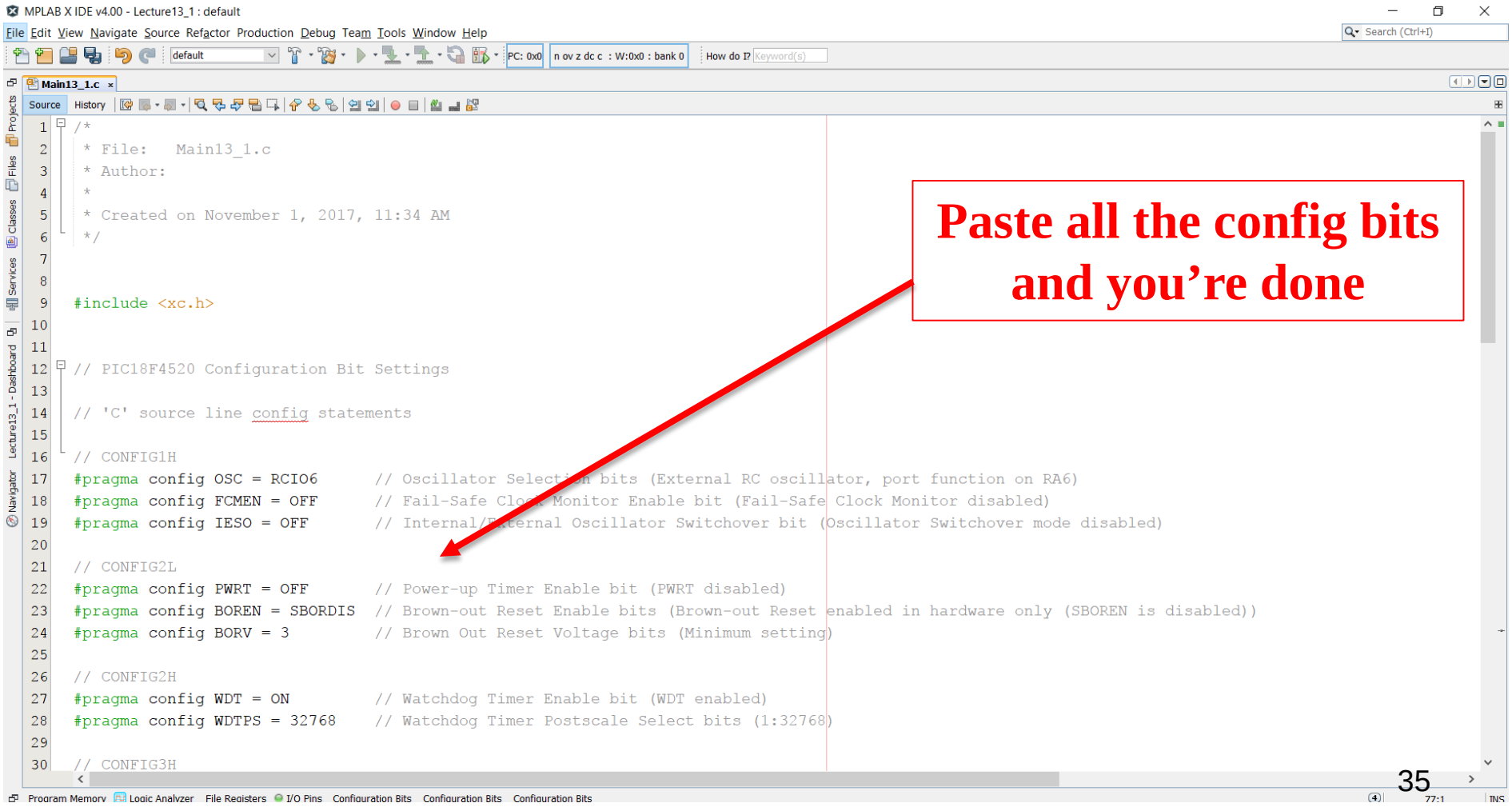
Config Bit Settings
MPLAB X IDE



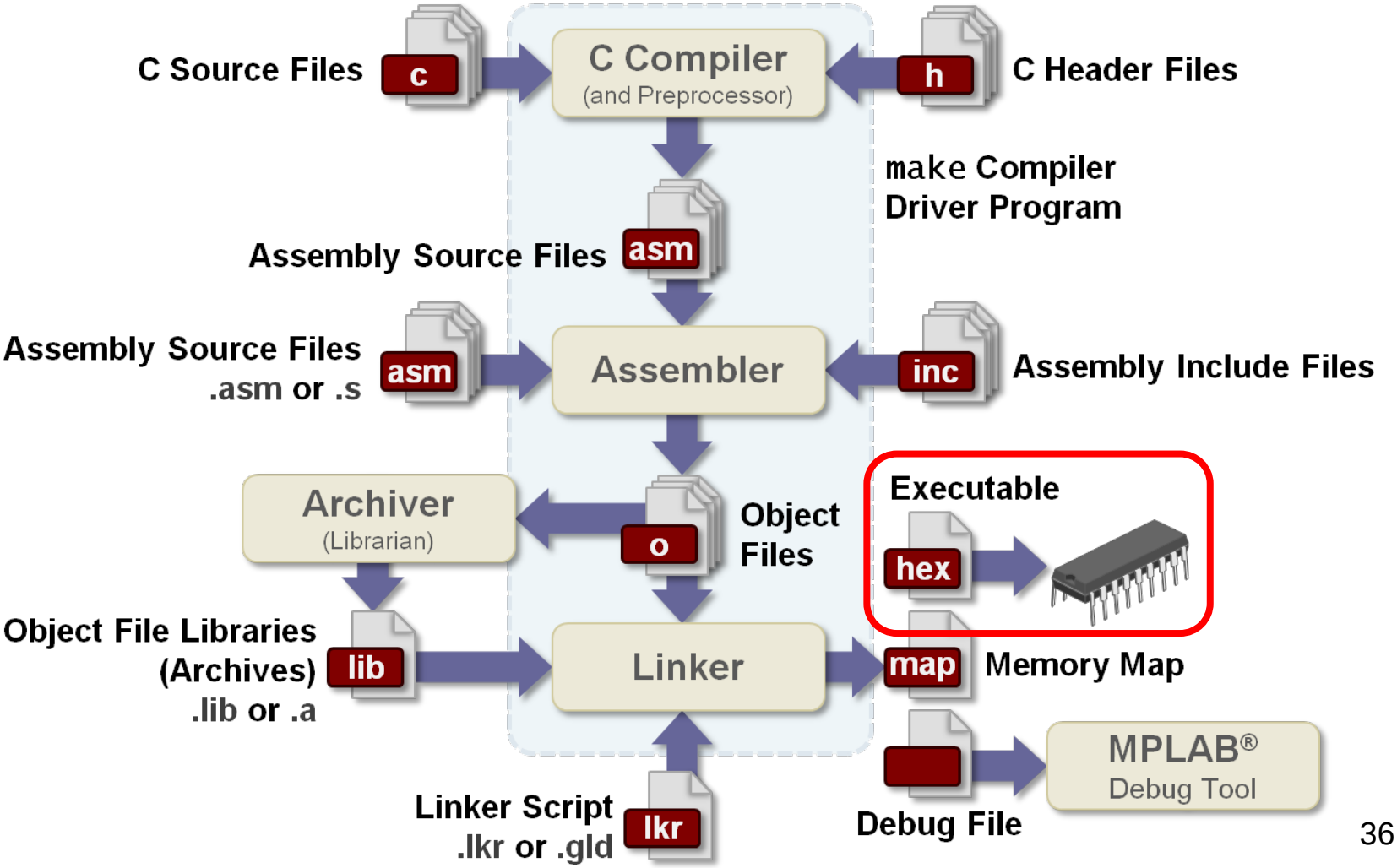
Config Bit Settings
MPLAB X IDE



Config Bit Settings
MPLAB X IDE



Software Program Flow




```
:0600000000F03CEF04F0EB
:1006000080202020202020202000C020202020200A
:10061000202020200080456D625379735F3100C05007
:1006200049437265736574003332280 0C0600007B
:100630000000FFFF11EC04F0929692949292809643
:10064000809480920 0E086E000E076EC2EC03F0D6
:10065000060E086E0A0E076EC2EC03F0060E086E58
:10066000140E076EC2EC03F0060E086E1E0E076E27
:10067000C2EC03F0FFFF80760C6E640E016E0C502E
:1006800028EC04F00C6E640E016E0C5028EC04F0A3
:1006900080740C6E640E016E0C5028EC04F00C6E2D
:1006A000640E016E0C5028EC04F080720C6E640E27
:1006B000016E0C5 C6E640E016E0C50B0
:1006C00028EC0 0C6E64 E0C5028EC04F063
:1006D0000C6E 16E0C 04F00C6E640E6F
:1006E000016E 8EC04 640E016E0C5080
:1006F00028EC04F00C6E64 0C5028EC04F033
:100700000C6E640E016E0 EC04F0B3D7FFFFA2
:10071000076E0A0E0D6E 04D0FFFFECEC03F0DD
:100720000D06FFFF0D6 9D78490060E096E280EA0
:10073000086E1AD0FFF 08C0F6FF09C0F7FF0800D7
:10074000FFFFF5CF0AF 084820AC083FF8492ECECAD
:1007500003F00A3AF00E0A1684820AC083FF8492DC
:10076000ECEC03F00 02AFFFF08C0F6FF09C0B5
:10077000F7FF0800 500009D8B41200FFFF93
:10078000D9D7FFFF 9019D0FFFF07C0F6FF06
:1007900008C0F7FF0800FFFFF5CF09F0848209C009
:1007A00083FF8492093AF00E0916848209C083FF00
:1007B0008492ECEC03F08480074A082AFFFF07C00C
:1007C000F6FF08C0F7FF0800FFFFF5500009D8B496
:1007D0001200FFFFDAD7FFFFFFFF000E046E020ECC
:1007E000036E000E066E000E056E000E066E000E05
:1007F000056E09D0FFFF026EE20E016E025032EC70
:1008000004F0054A062AFFFF0650800A026E0350D4
:10081000055C0450800A0258D8B01200FFFFEAD7E6
:10082000FFFFFFFF8E0EC16EE10E926EDC0E936E27
:10083000D00E946E0F0E956E000E966E100E806E9A
:100840000B6E000E0E6E0B5087EC03F01200FFFFD4
:10085000FFFFFFFFFA0EFFFFE82EFDD7012EF9D7AD
:100860001200FFFFFFFFFFFF190EFFFFE82EFDD76D
:10087000012EF9D71200FFFF0E6A000EF86E00017C
:0408800019EF03F079
:020000040020DA
:08000000FFFFFFFFFFFFFFFF00
:020000040030CA
:0E000000FF220D0EFF0181FF0FC00FE00F4029
:00000001FF
```

```
                  00010 R1  equ 0x07
                  00011 R2  equ 0x08
                  00012 R3  equ 0x09
```

OPCODE

```
00000             00014       ORG 0
00000 6A93        00015       CLRF  TRISB
00002 0E55        00016       MOVLW 0x55
00004 6E81        00017       MOVWF PORTB
00006 1E81        00018 L3    COMF  PORTB,F
00008 EC78 F094   00019       CALL  QDELAY
0000C D7FC        00020       BRA   L3

                  00023 ;-----------1/4 SECO
128F0             00024       ORG 128F0H
128F0             00025 QDELAY
128F0 0E02        00026       MOVLW D'2'
128F2 6E07        00027       MOVWF R1
128F4 0EFA        00028 D1    MOVLW D'250'
128F6 6E08        00029       MOVWF R2
128F8 0EFA        00030 D2    MOVLW D'250'
128FA 6E09        00031       MOVWF R3
128FC 0000        00032 D3    NOP
128FE 0000        00033       NOP
2900 0609         00034       DECF  R3, F
2902 E1FC         00035       BNZ   D3
2904 0608         00036       DECF  R2, F
2906 E1F8         00037       BNZ   D2
2908 0607         00038       DECF  R1, F
290A E1F4         00039       BNZ   D1
290C 0012         00040       RETURN
                  00041       END
```

MOVLW	**Move literal to W**
Syntax:	[*label*] MOVLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → W
Status Affected:	None
Encoding:	0000 | 1110 | kkkk | kkkk

0b1110 = 0xE

MOVWF	**Move W to f**
Syntax:	[*label*] MOVWF f [,a]
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]
Operation:	(W) → f
Status Affected:	None
Encoding:	0110 | 111a | ffff | ffff

0b0110 111a = 0x6E or 0x6F

So, What's in a HEX File?

Table 8-13: Intel Hex File Formats Produced by MPLAB (See http://www.microchip.com)

Format Name	Format Type	File Extension	Max. ROM Address
Intel Hex format	INHX8M	.hex	16-bit address
Intel Hex 32 format	INHX32	.hex	32-bit address
Intel Split Hex	INHX8S	.hxl and .hxh	16-bit address for each

```
:10000000936A550E816E811E07EC00F0FCD7020E3C
:10001000076EFA0E086EFA0E096E0000000009065F
:0C002000FCE10806F8E10706F4E112001C
:0300010022020ECA
:010006008079
:060008000FC00FE00F40E5
:00000001FF
```

INHX8M

Separating the fields, we get the following:

```
:BB AAAA TT HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH          CC
:10 0000 00 936A550E816E811E07EC00F0FCD7020E          3C
:10 0010 00 076EFA0E086EFA0E096E0000000009065F        5F
:0C 0020 00 FCE10806F8E10706F4E112001C                1C

:03 0001 00 22020E                                    CA
:01 0006 00 80                                        79
:06 0008 00 0FC00FE00F40                              E5
:00 0000 01                                           FF
```

```
:020000040000FA
:0E000000936A550E816E811E78EC94F0FCD749
:020000040001F9
:1028F000020E076EFA0E086EFA0E096E0000000056
:0E2900000906FCE10806F8E10706F4E1120002
:020000040030CA
:0600010022020E830180C3
:06000800FFC0FFE0FF4015
:00000001FF
```

INHX32

Separating the fields we get the following:

```
:BB AAAA TT HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH          CC
:02 0000 04 0000                                      FA
:0E 0000 00 936A550E816E811E78EC94F0FCD7               49
:02 0000 04 0001                                      F9
:10 28F0 00 020E076EFA0E086EFA0E096E00000000          56
:0E 2900 00 0906FCE10806F8E10706F4E11200              02
:02 0000 04 0030                                      CA
:06 0001 00 22020E830180                              C3
:06 0008 00 FFC0FFE0FF40                              15
:00 0000 01                                           FF
```

HEX File Content
INHX8M

```
31 :1007D0001200FFFFDAD7FFFFFFFF000E046E020ECC
32 :1007E000036E000E066E000E056E000E066E000E05
33 :1007F000056E09D0FFFF026EE20E016E025032EC70
34 :1008000004F0054A062AFFFF0650800A026E0350D4
35 :10081000055C0450800A0258D8B01200FFFFEAD7E6
36 :10082000FFFFFFFF8E0EC16EE10E926EDC0E936E27
37 :10083000D00E946E0F0E956E000E966E100E806E9A
38 :100840000B6E000E0E6E0B5087EC03F01200FFFFD4
```

BB (count byte): how many DATA bytes in the line, max value 0x10 (16)

AAAA (address): ROM address space where data needs to go

TT (type): 00 means hex file is continuing, 01 means it is over (EOF)

HHH…HHH (data): opcode and data/location, at most 16 in a line – see BB

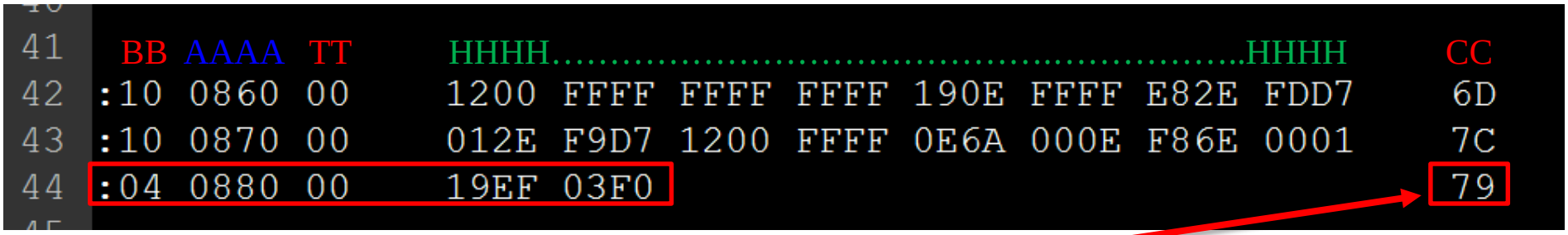
CC (checksum): to ensure integrity of incoming hex code

```
41   BB AAAA TT     HHHH.....................................HHHH     CC
42 :10 0860 00     1200 FFFF FFFF FFFF 190E FFFF E82E FDD7     6D
43 :10 0870 00     012E F9D7 1200 FFFF 0E6A 000E F86E 0001     7C
44 :04 0880 00     19EF 03F0                                   79
45
```

Checksum

- **Calculate Checksum Byte**
  1. Add all bytes together and drop the carries (mask anything above 8 bits)
  2. Take the 2's complement of the sum, and that is now your checksum byte
  3. Append as the last byte in the series in the HEX file row

- **Check Integrity** (perform checksum operation)
  1. Add all bytes together with checksum byte
     - If bottom 8 bits == ZERO → NO CORRUPTION
     - If bottom 8 bits != ZERO → SOMETHING'S WRONG



- **Calculate checksum byte**
  1. Add all bytes together and drop the carries (mask anything above 8 bits)
  2. Take the 2's complement of the sum, and that is now your checksum byte
  3. Append as the last byte in the series in the HEX file row

Data to Send	
0x04	
0x08	
0x80	
0x00	+
0x19	
0xEF	
0x03	
0xF0	
0x287 = 0b 10 1000 0111	

```
41   BB AAAA TT     HHHH.....................................HHHH     CC
42  :10 0860 00     1200 FFFF FFFF FFFF 190E FFFF E82E FDD7     6D
43  :10 0870 00     012E F9D7 1200 FFFF 0E6A 000E F86E 0001     7C
44  :04 0880 00     19EF 03F0                                   79
```

- **Calculate checksum byte**
  1. Add all bytes together and drop the carries (mask anything above 8 bits)
  2. Take the 2's complement of the sum, and that is now your checksum byte
  3. Append as the last byte in the series in the HEX file row

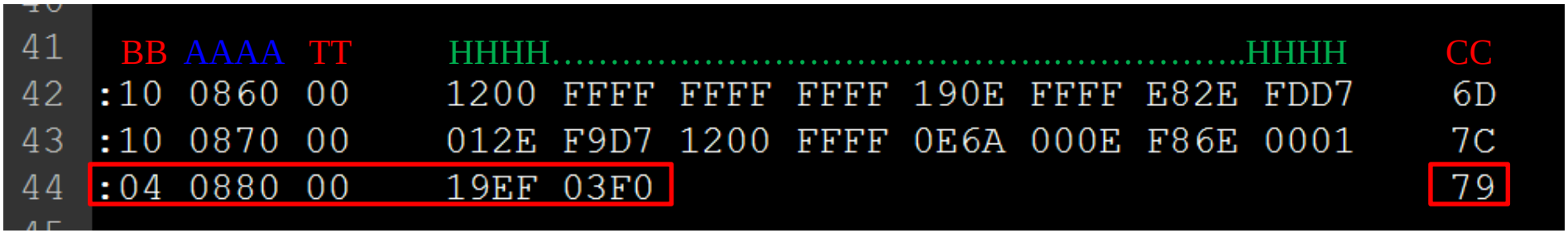
Data to Send

	0x04
	0x08
	0x80
+	0x00
	0x19
	0xEF
	0x03
	0xF0
0x287 = 0b 10	1000 0111

Take 2's Complement

0x87 = 0b	1000 0111
	1000 0111
Invert	0111 1000
Add 1	+ 1
	0111 1001
	= 0x79

0x79 is your Checksum Byte

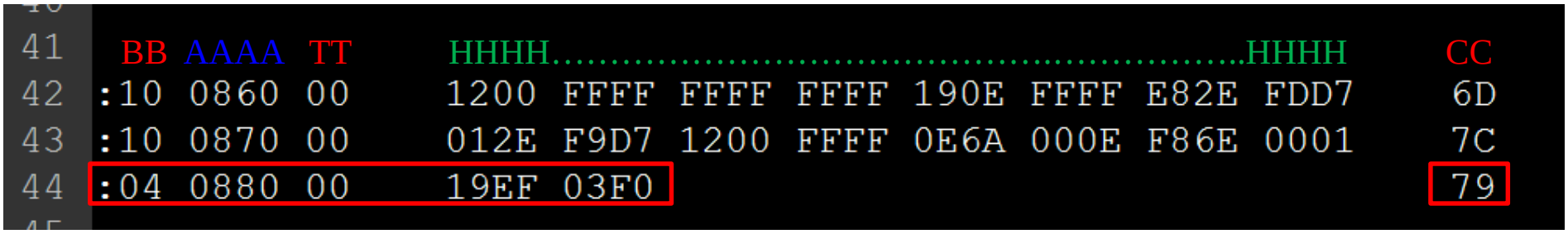


Check integrity (perform checksum operation)

1. Add all bytes together with checksum byte **(0x79)**
   - If bottom 8 bits == ZERO → NO CORRUPTION
   - If bottom 8 bits != ZERO → SOMETHING'S WRONG

Data Sent (truth)
0x04
0x08
0x80
0x00
0x19
0xEF
0x03
0xF0

Data Received (GOOD)	
0x04	
0x08	
0x80	
+	0x00
0x19	
0xEF	
0x03	
0xF0	
0x79	
0x300 = 0b 11	0000 0000



Check integrity (perform checksum operation)

1. Add all bytes together with checksum byte **(0x79)**
   - If bottom 8 bits == ZERO → NO CORRUPTION
   - If bottom 8 bits != ZERO → SOMETHING'S WRONG

Data Sent (truth)

0x04
0x08
0x80
0x00
0x19
0xEF
0x03
0xF0

Data Received (GOOD)

0x04
0x08
0x80
\+ 0x00
0x19
0xEF
0x03
0xF0
0x79

0x300 = 0b 11 0000 0000

Data Received (BAD)

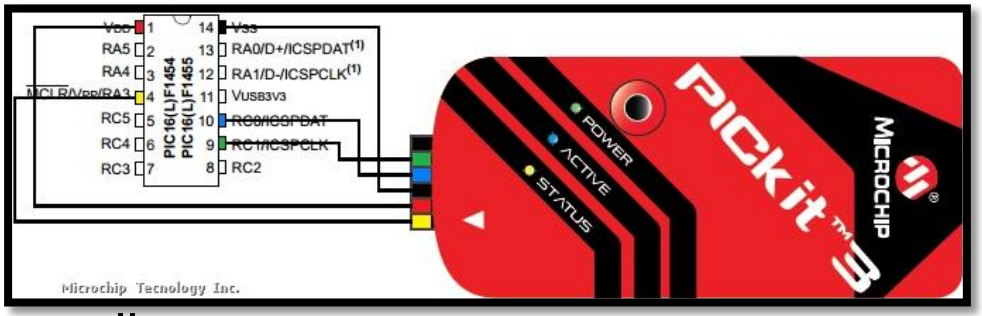
0x04
0x09
0x80
\+ 0x00
0x19
0xEF
0x03
0xF0
0x79

0x301 = 0b 11 0000 0001

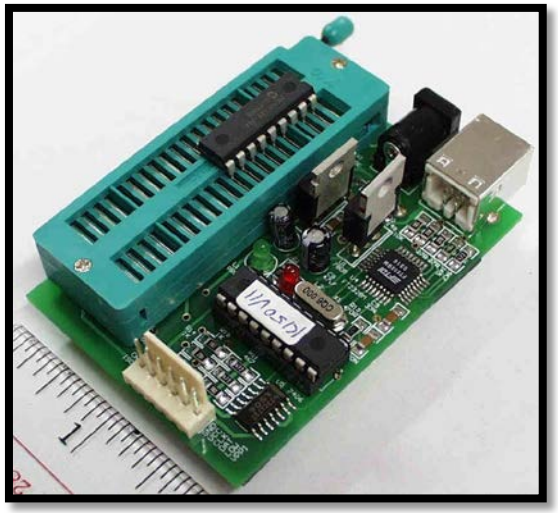
Can't trust data

We have the Hex File, What Now?

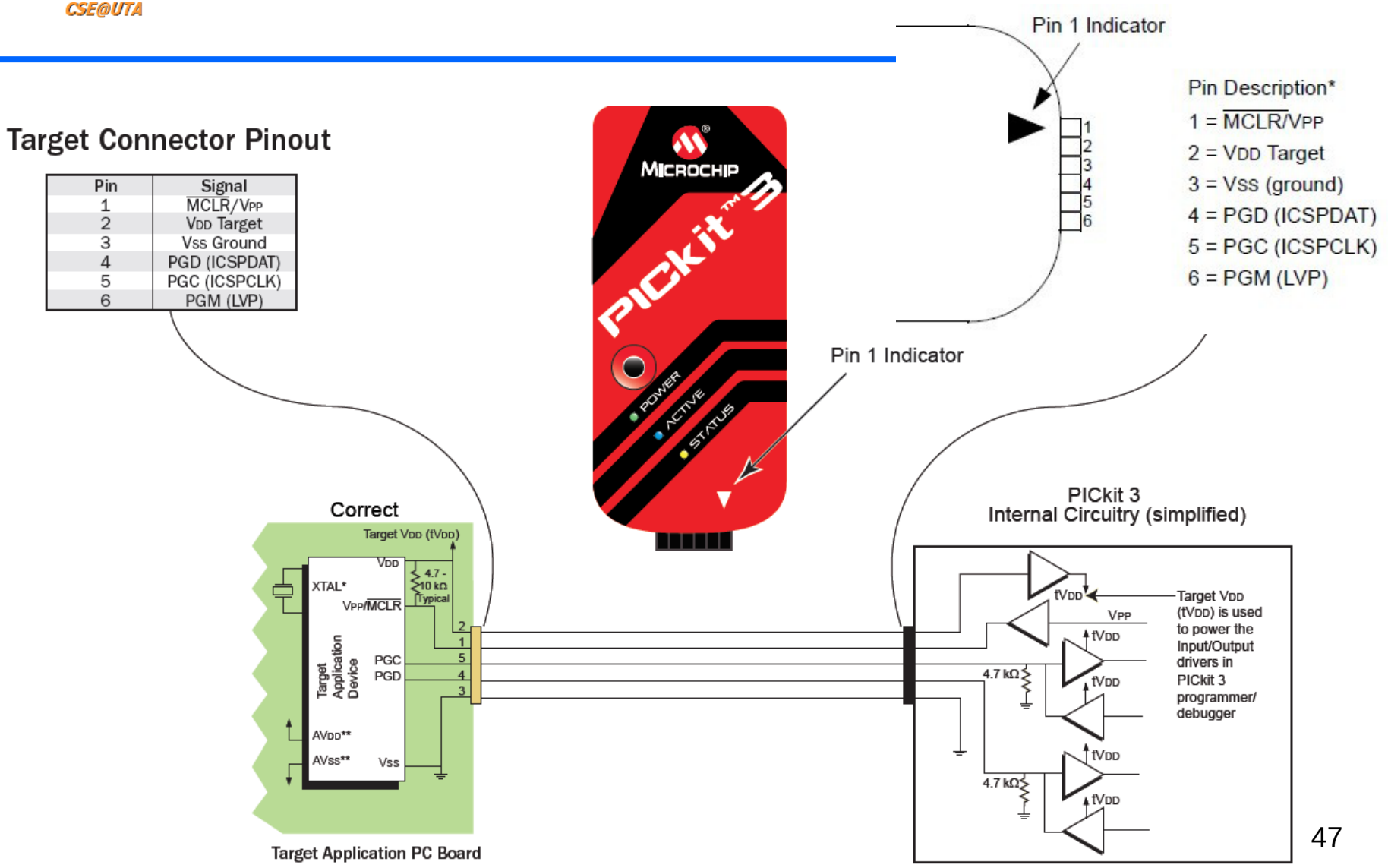
- Need to burn/flash the .hex file onto the microcontroller



- Three methods:
  - **Off-circuit (OCSP)**: the microcontroller is programmed as a stand alone chip and is then inserted into the circuit
  - **In-circuit (ICSP)**: microcontroller has to set aside pins that are used to program it while inside the circuit
  - **Boot loader**: a special code running on the microcontroller, allowing it to accept code from any of its interfaces (boot loader needs to be burned with one of the previous two methods beforehand)



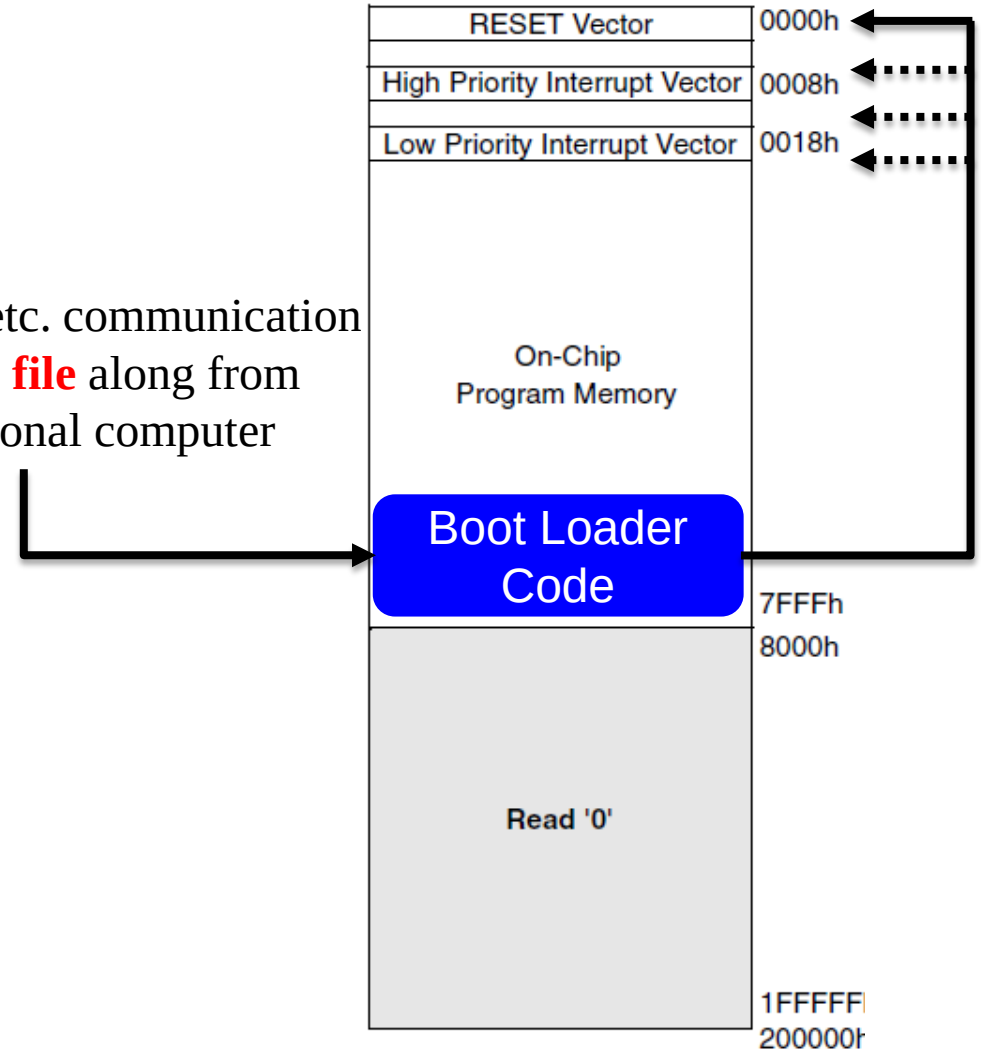
PicKit 3 - ICSP



Boot loader Pros/Cons

- See our QwikFlash boards (old method)
- Takes away ROM space from the developer
- Don’t need an actual programmer (PICKit)
- Can use any communication methods available to the microcontroller (SPI, I2C, etc.)
  - But slower
- Code space for boot loader must be reserved and protected
- May be a good choice for development but usually not for final product

Boot Loader in Program ROM



Use Serial/I2C/etc. communication to send **HEX file** along from regular personal computer

Use **existing program code** to insert your **new program code**

Questions?