Welcome to E-XFL.COM

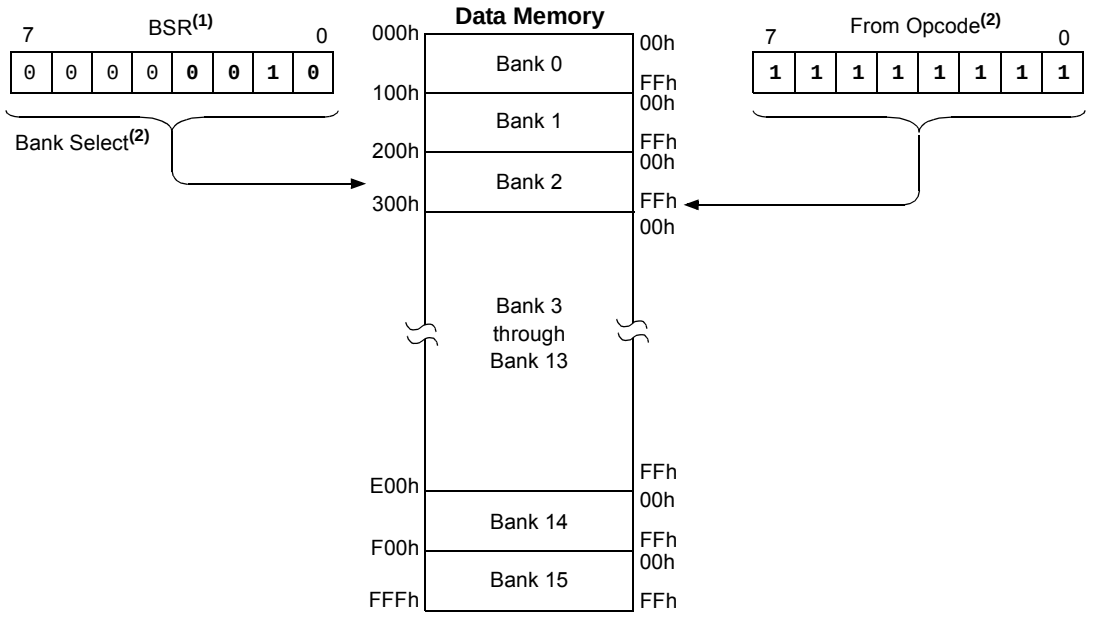
What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LCD, POR, PWM, WDT
Number of I/O	69
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 24x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f86k90-i-ptrsl

FIGURE 6-7: USE OF THE BANK SELECT REGISTER (DIRECT ADDRESSING)



Note 1: The Access RAM bit of the instruction can be used to force an override of the selected bank (BSR<3:0>) to the registers of the Access Bank.

2: The MOVFF instruction embeds the entire 12-bit address in the instruction.

6.3.2 ACCESS BANK

While the use of the BSR, with an embedded 8-bit address, allows users to address the entire range of data memory, it also means that the user must ensure that the correct bank is selected. If not, data may be read from, or written to, the wrong location. This can be disastrous if a GPR is the intended target of an operation, but an SFR is written to instead. But verifying and/or changing the BSR for each read or write to data memory can become very inefficient.

To streamline access for the most commonly used data memory locations, the data memory is configured with an Access Bank, which allows users to access a mapped block of memory without specifying a BSR. The Access Bank consists of the first 96 bytes of memory (00h-5Fh) in Bank 0 and the last 160 bytes of memory (60h-FFh) in Bank 15. The lower half is known as the “Access RAM” and is composed of GPRs. The upper half is where the device’s SFRs are mapped. These two areas are mapped contiguously in the Access Bank and can be addressed in a linear fashion by an 8-bit address (Figure 6-6).

The Access Bank is used by core PIC18 instructions that include the Access RAM bit (the ‘a’ parameter in the instruction). When ‘a’ is equal to ‘1’, the instruction uses the BSR and the 8-bit address included in the opcode for the data memory address. When ‘a’ is ‘0’, however, the instruction is forced to use the Access Bank address map. In that case, the current value of the BSR is ignored entirely.

Using this “forced” addressing allows the instruction to operate on a data address in a single cycle without updating the BSR first. For 8-bit addresses of 60h and above, this means that users can evaluate and operate on SFRs more efficiently. The Access RAM below 60h is a good place for data values that the user might need to access rapidly, such as immediate computational results or common program variables.

Access RAM also allows for faster and more code efficient context saving and switching of variables.

The mapping of the Access Bank is slightly different when the extended instruction set is enabled (XINST Configuration bit = 1). This is discussed in more detail in **Section 6.6.3 “Mapping the Access Bank in Indexed Literal Offset Mode”**.

6.3.3 GENERAL PURPOSE REGISTER FILE

PIC18 devices may have banked memory in the GPR area. This is data RAM which is available for use by all instructions. GPRs start at the bottom of Bank 0 (address 000h) and grow upwards towards the bottom of the SFR area. GPRs are not initialized by a Power-on Reset and are unchanged on all other Resets.

NOTES:

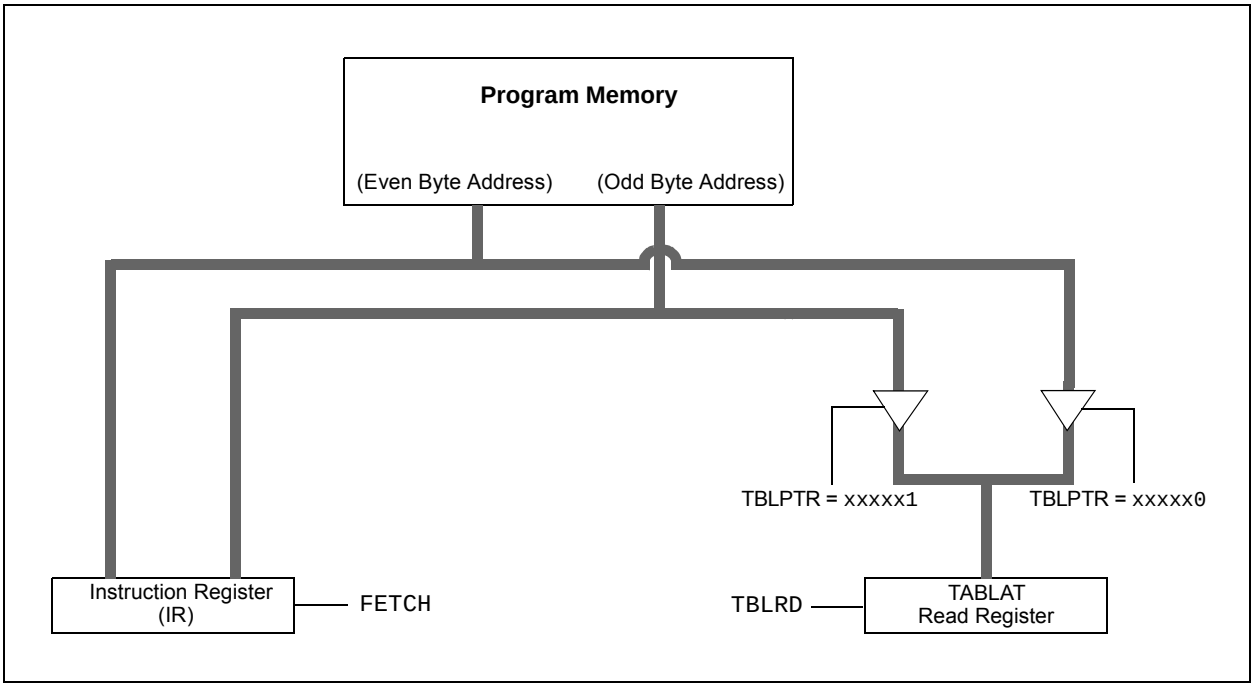
7.3 Reading the Flash Program Memory

The TBLRD instruction is used to retrieve data from program memory and places it into data RAM. Table reads from program memory are performed one byte at a time.

The TBLPTR points to a byte address in program memory space. Executing TBLRD places the byte pointed to into TABLAT. In addition, TBLPTR can be modified automatically for the next table read operation.

The internal program memory is typically organized by words. The Least Significant bit of the address selects between the high and low bytes of the word. Figure 7-4 shows the interface between the internal program memory and the TABLAT.

FIGURE 7-4: READS FROM FLASH PROGRAM MEMORY



EXAMPLE 7-1: READING A FLASH PROGRAM MEMORY WORD

```
            BCF     EECON1, CFGS          ; point to Flash program memory
            BSF     EECON1, EEPGD         ; access Flash program memory
            MOVLW   CODE_ADDR_UPPER       ; Load TBLPTR with the base
            MOVWF   TBLPTRU               ; address of the word
            MOVLW   CODE_ADDR_HIGH
            MOVWF   TBLPTRH
            MOVLW   CODE_ADDR_LOW
            MOVWF   TBLPTRL
READ_WORD
            TBLRD*+                       ; read into TABLAT and increment
            MOVF    TABLAT, W             ; get data
            MOVWF   WORD_EVEN
            TBLRD*+                       ; read into TABLAT and increment
            MOVF    TABLAT, W             ; get data
            MOVF    WORD_ODD
```

8.3 Reading the Data EEPROM Memory

To read a data memory location, the user must write the address to the EEADRH:EEADR register pair, clear the EEPGD control bit (EECON1<7>) and then set control bit, RD (EECON1<0>). After one cycle, the data is available in the EEDATA register; therefore, it can be read after one NOP instruction. EEDATA will hold this value until another read operation, or until it is written to by the user (during a write operation). The basic process is shown in Example 8-1.

8.4 Writing to the Data EEPROM Memory

To write an EEPROM data location, the address must first be written to the EEADRH:EEADR register pair and the data written to the EEDATA register. The sequence in Example 8-2 must be followed to initiate the write cycle.

The write will not begin if this sequence is not exactly followed (write 0x55 to EECON2, write 0xAA to EECON2, then set WR bit) for each byte. It is strongly recommended that interrupts be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable writes. This mechanism prevents accidental writes to data EEPROM due to unexpected code execution (i.e., runaway programs). The WREN bit should be kept clear at all times, except when updating the EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, EECON1, EEADRH:EEADR and EEDATA cannot be modified. The WR bit will be inhibited from being set unless the WREN bit is set. The WREN bit must be set on a previous instruction. Both WR and WREN cannot be set with the same instruction.

At the completion of the write cycle, the WR bit is cleared in hardware and the EEPROM Interrupt Flag bit (EEIF) is set. The user may either enable this interrupt, or poll this bit. EEIF must be cleared by software.

8.5 Write Verify

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

Note:	Self-write execution to Flash and EEPROM memory cannot be done while running in LP Oscillator mode (Low-Power mode). Therefore, executing a self-write will put the device into High-Power mode.

EXAMPLE 8-1: DATA EEPROM READ

```
        MOVLW   DATA_EE_ADDRH      ;
        MOVWF   EEADRH             ; Upper bits of Data Memory Address to read
        MOVLW   DATA_EE_ADDR       ;
        MOVWF   EEADR              ; Lower bits of Data Memory Address to read
        BCF     EECON1, EEPGD      ; Point to DATA memory
        BCF     EECON1, CFGS       ; Access EEPROM
        BSF     EECON1, RD         ; EEPROM Read
        NOP
        MOVF    EEDATA, W          ; W = EEDATA
```

EXAMPLE 8-2: DATA EEPROM WRITE

```
                MOVLW   DATA_EE_ADDRH      ;
                MOVWF   EEADRH             ; Upper bits of Data Memory Address to write
                MOVLW   DATA_EE_ADDR       ;
                MOVWF   EEADR              ; Lower bits of Data Memory Address to write
                MOVLW   DATA_EE_DATA       ;
                MOVWF   EEDATA             ; Data Memory Value to write
                BCF     EECON1, EEPGD      ; Point to DATA memory
                BCF     EECON1, CFGS       ; Access EEPROM
                BSF     EECON1, WREN       ; Enable writes

                BCF     INTCON, GIE        ; Disable Interrupts
                MOVLW   0x55               ;
  Required      MOVWF   EECON2             ; Write 55h
  Sequence      MOVLW   0xAA               ;
                MOVWF   EECON2             ; Write 0AAh
                BTFSC   EECON1, WR         ; Wait for write to complete
                GOTO    $-2
                BSF     INTCON, GIE        ; Enable Interrupts

                                           ; User code execution
                BCF     EECON1, WREN       ; Disable writes on write complete (EEIF set)
```

TABLE 8-1: REGISTERS ASSOCIATED WITH DATA EEPROM MEMORY

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	75
EEADRH	—	—	—	—	—	—	EEPROM Address Register High Byte		79
EEADR	EEPROM Address Register Low Byte								80
EEDATA	EEPROM Data Register								80
EECON2	EEPROM Control Register 2 (not a physical register)								79
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	79

Legend: — = unimplemented, read as '0'. Shaded cells are not used during Flash/EEPROM access.

REGISTER 10-7: PIR4: PERIPHERAL INTERRUPT FLAG REGISTER 4

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CCP10IF(1)	CCP9IF(1)	CCP8IF	CCP7IF	CCP6IF	CCP5IF	CCP4IF	CCP3IF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-1 **CCP10IF:CCP4IF:** CCP<10:4> Interrupt Flag bits(1)

Capture Mode
1 = A TMR register capture occurred (must be cleared in software)
0 = No TMR register capture occurred

Compare Mode
1 = A TMR register compare match occurred (must be cleared in software)
0 = No TMR register compare match occurred

PWM Mode
Not used in PWM mode.

bit 0 **CCP3IF:** ECCP3 Interrupt Flag bits

Capture Mode
1 = A TMR register capture occurred (must be cleared in software)
0 = No TMR register capture occurred

Compare Mode
1 = A TMR register compare match occurred (must be cleared in software)
0 = No TMR register compare match occurred

PWM Mode
Not used in PWM mode.

Note 1: Unimplemented in devices with a program memory of 32 Kbytes (PIC18FX5K90).

TABLE 11-1: PORTA FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RA0/AN0/ULPWU	RA0	0	O	DIG	LATA<0> data output; not affected by analog input.
		1	I	TTL	PORTA<0> data input; disabled when analog input is enabled.
	AN0	1	I	ANA	A/D Input Channel 0. Default input configuration on POR; does not affect digital output.
	ULPWU	1	I	ANA	Ultra Low-Power Wake-up (ULPWU) input.
RA1/AN1/SEG18	RA1	0	O	DIG	LATA<1> data output; not affected by analog input.
		1	I	TTL	PORTA<1> data input; disabled when analog input is enabled.
	AN1	1	I	ANA	A/D Input Channel 1. Default input configuration on POR; does not affect digital output.
	SEG18	1	O	ANA	LCD Segment 18 output; disables all other pin functions.
RA2/AN2/VREF-	RA2	0	O	DIG	LATA<2> data output; not affected by analog input.
		1	I	TTL	PORTA<2> data input; disabled when analog functions are enabled.
	AN2	1	I	ANA	A/D Input Channel 2. Default input configuration on POR.
	VREF-	1	I	ANA	A/D and comparator low reference voltage input.
RA3/AN3/VREF+	RA3	0	O	DIG	LATA<3> data output; not affected by analog input.
		1	I	TTL	PORTA<3> data input; disabled when analog input is enabled.
	AN3	1	I	ANA	A/D Input Channel 3. Default input configuration on POR.
	VREF+	1	I	ANA	A/D and comparator high reference voltage input.
RA4/T0CKI/ SEG14	RA4	0	O	DIG	LATA<4> data output.
		1	I	ST	PORTA<4> data input. Default configuration on POR.
	T0CKI	x	I	ST	Timer0 clock input.
	SEG14	1	O	ANA	LCD Segment 14 output; disables all other pin functions.
RA5/AN4/SEG15/ T1CKI/T3G/ HLVDIN	RA5	0	O	DIG	LATA<5> data output; not affected by analog input.
		1	I	TTL	PORTA<5> data input; disabled when analog input is enabled.
	AN4	1	I	ANA	A/D Input Channel 4. Default configuration on POR.
	SEG15	1	O	ANA	LCD Segment 15 output; disables all other pin functions.
	T1CKI	x	I	ST	Timer1 clock input.
	T3G	x	I	ST	Timer3 external clock gate input.
	HLVDIN	1	I	ANA	High/Low-Voltage Detect (HLVD) external trip point input.
OSC2/CLKO/RA6	OSC2	x	O	ANA	Main oscillator feedback output connection (HS, XT and LP modes).
	CLKO	x	O	DIG	System cycle clock output (FOSC/4, EC and INTOSC modes).
	RA6	0	O	DIG	LATA<6> data output; disabled when OSC2 Configuration bit is set.
		1	I	TTL	PORTA<6> data input; disabled when OSC2 Configuration bit is set.
OSC1/CLKI/RA7	OSC1	x	I	ANA	Main oscillator input connection (HS, XT and LP modes).
	CLKI	x	I	ANA	Main external clock source input (EC modes).
	RA7	0	O	DIG	LATA<7> data output; disabled when OSC2 Configuration bit is set.
		1	I	TTL	PORTA<7> data input; disabled when OSC2 Configuration bit is set.

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Trigger Buffer Input, TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

TABLE 11-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	78
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	78
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	78
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	75
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	75
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	75
LCDSE1	SE15	SE14	SE13	SE12	SE11	SE10	SE09	SE08	83
LCDSE3	SE31	SE30	SE29	SE28	SE27	SE26	SE25	SE24	83

Legend: Shaded cells are not used by PORTB.

TABLE 11-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	78
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	78
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	78
LCDSE0	SE07	SE06	SE05	SE04	SE03	SE02	SE01	SE00	83
PADCFG1	RDPU	REPU	RJPU(1)	—	—	RTSECSEL1	RTSECSEL0	—	80

Legend: Shaded cells are not used by PORTD.

Note 1: This bit is not available in 64-pin devices.

TABLE 11-10: PORTE FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RE0/LCDBIAS1/ P2D	RE0	0	O	DIG	LATE<0> data output.
		1	I	ST	PORTE<0> data input.
	LCDBIAS1	—	I	ANA	LCD module bias voltage input.
	P2D	0	O	—	ECCP2 PWM Output D. May be configured for tri-state during Enhanced PWM shutdown events.
RE1/LCDBIAS2/ P2C	RE1	0	O	DIG	LATE<1> data output.
		1	I	ST	PORTE<1> data input.
	LCDBIAS2	—	I	ANA	LCD module bias voltage input.
	P2C	0	O	—	ECCP2 PWM Output C. May be configured for tri-state during Enhanced PWM shutdown events.
RE2/LCDBIAS3/ P2B	RE2	0	O	DIG	LATE<2> data output.
		1	I	ST	PORTE<2> data input.
	LCDBIAS3	x	I	ANA	LCD module bias voltage input.
	P2B	0	O	—	ECCP2 PWM Output B. May be configured for tri-state during Enhanced PWM shutdown events.
RE3/COM0/ P3C/CCP9/ REFO	RE3	0	O	DIG	LATE<3> data output.
		1	I	ST	PORTE<3> data input.
	COM0	x	O	ANA	LCD Common 0 output; disables all other outputs.
	P3C	0	O	—	ECCP3 PWM Output C. May be configured for tri-state during Enhanced PWM shutdown events.
	CCP9(2)	0	O	DIG	CCP9 compare/PWM output; takes priority over port data.
		1	I	ST	CCP9 capture input.
	REFO	x	O	DIG	Reference output clock.
RE4/COM1/ P3B/CCP8	RE4	0	O	DIG	LATE<4> data output.
		1	I	ST	PORTE<4> data input.
	COM1	x	O	ANA	LCD Common 1 output; disables all other outputs.
	P3B	0	O	—	ECCP3 PWM Output B. May be configured for tri-state during Enhanced PWM shutdown events.
	CCP8	0	O	DIG	CCP8 Compare/PWM output; takes priority over port data.
		1	I	ST	CCP8 capture input.
RE5/COM2/ P1C/CCP7	RE5	0	O	DIG	LATE<5> data output.
		1	I	ST	PORTE<5> data input.
	COM2	x	O	ANA	LCD Common 2 output; disables all other outputs.
	P1C	0	O	—	ECCP1 PWM Output C. May be configured for tri-state during Enhanced PWM shutdown events.
	CCP7	0	O	DIG	CCP7 Compare/PWM output; takes priority over port data.
		1	I	ST	CCP7 capture input.
RE6/COM3/ P1B/CCP6	RE6	0	O	DIG	LATE<6> data output.
		1	I	ST	PORTE<6> data input.
	COM3	x	O	ANA	LCD Common 3 output; disables all other outputs.
	P1B	0	O	—	ECCP1 PWM Output B. May be configured for tri-state during Enhanced PWM shutdown events.
	CCP6	0	O	DIG	CCP6 Compare/PWM output; takes priority over port data.
		1	I	ST	CCP6 capture input.

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Trigger Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: Alternate assignment for ECCP2 when the CCP2MX Configuration bit is cleared.

2: This bit is unimplemented in PIC18FX5K90 devices.

NOTES:

REGISTER 17-18: ALRMMIN: ALARM MINUTES VALUE REGISTER

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

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

bit 6-4 **MINTEN<2:0>:** Binary Coded Decimal Value of Minute's Tens Digit bits
Contains a value from 0 to 5.

bit 3-0 **MINONE<3:0>:** Binary Coded Decimal Value of Minute's Ones Digit bits
Contains a value from 0 to 9.

REGISTER 17-19: ALRMSEC: ALARM SECONDS VALUE REGISTER

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

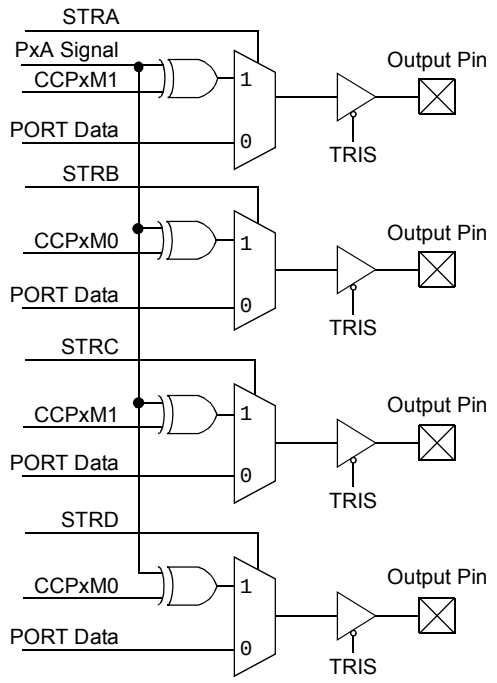
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

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

bit 6-4 **SECTEN<2:0>:** Binary Coded Decimal Value of Second's Tens Digit bits
Contains a value from 0 to 5.

bit 3-0 **SECONE<3:0>:** Binary Coded Decimal Value of Second's Ones Digit bits
Contains a value from 0 to 9.

FIGURE 19-16: SIMPLIFIED STEERING BLOCK DIAGRAM



Note 1: PORT outputs are configured as displayed when the CCPxCON register bits, PxM<1:0> = 00 and CCP1Mx<3:2> = 11.

2: Single PWM output requires setting at least one of the STRx bits.

19.4.7.1 Steering Synchronization

The STRSYNC bit of the PSTRxCON register gives the user two choices for when the steering event will happen. When the STRSYNC bit is '0', the steering event will happen at the end of the instruction that writes to the PSTRxCON register. In this case, the output signal at the Px<D:A> pins may be an incomplete PWM waveform. This operation is useful when the user firmware needs to immediately remove a PWM signal from the pin.

When the STRSYNC bit is '1', the effective steering update will happen at the beginning of the next PWM period. In this case, steering on/off the PWM output will always produce a complete PWM waveform.

Figures 19-17 and 19-18 illustrate the timing diagrams of the PWM steering depending on the STRSYNC setting.

FIGURE 19-17: EXAMPLE OF STEERING EVENT AT END OF INSTRUCTION (STRSYNC = 0)

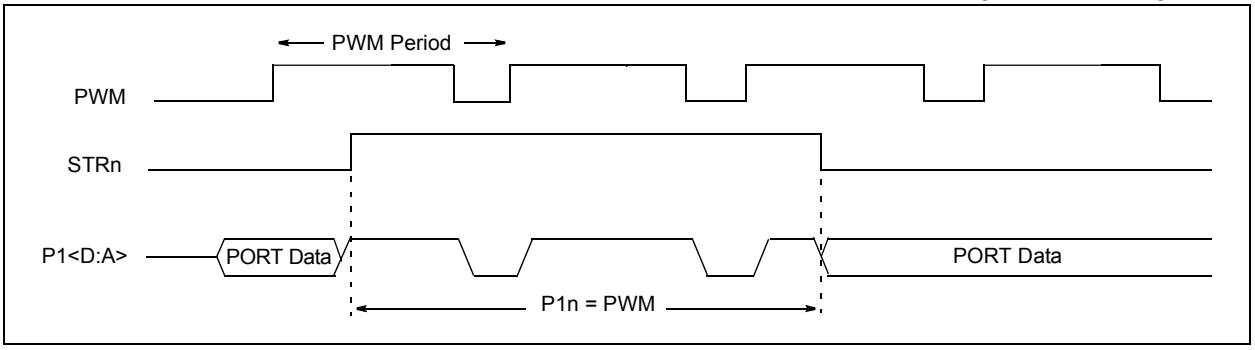
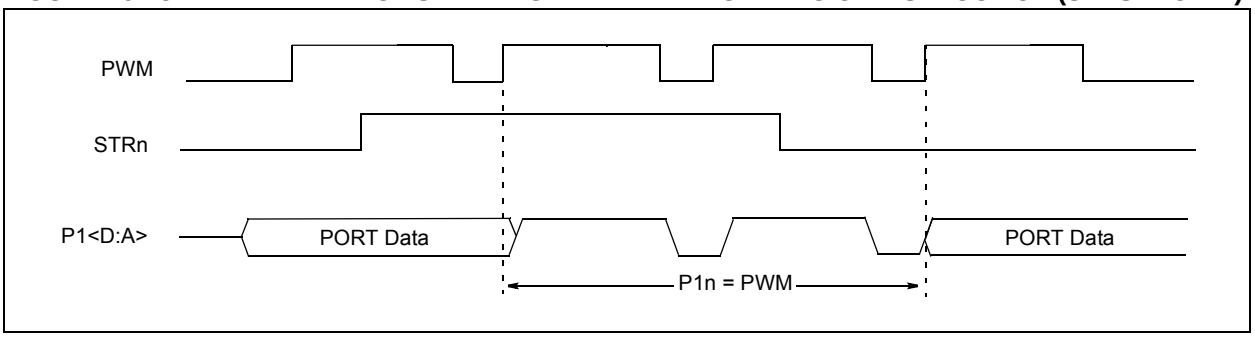


FIGURE 19-18: EXAMPLE OF STEERING EVENT AT BEGINNING OF INSTRUCTION (STRSYNC = 1)



20.7 LCD Frame Frequency

The rate at which the COM and SEG outputs change is called the LCD frame frequency.

TABLE 20-5: FRAME FREQUENCY FORMULAS

Multiplex	Frame Frequency =
Static	Clock Source/(4 x 1 x (LP<3:0> + 1))
1/2	Clock Source/(2 x 2 x (LP<3:0> + 1))
1/3	Clock Source/(1 x 3 x (LP<3:0> + 1))
1/4	Clock Source/(1 x 4 x (LP<3:0> + 1))

Note: Clock source is (FOSC/4)/8192, Timer1 Osc/32 or INTRC/32.

TABLE 20-6: APPROXIMATE FRAME FREQUENCY (IN Hz) USING FOSC AT 32 MHz, TIMER1 AT 32.768 kHz OR INTRC OSC

LP<3:0>	Static	1/2	1/3	1/4
1	125	125	167	125
2	83	83	111	83
3	62	62	83	62
4	50	50	67	50
5	42	42	56	42
6	36	36	48	36
7	31	31	42	31

20.8 LCD Waveform Generation

LCD waveform generation is based on the philosophy that the net AC voltage across the dark pixel should be maximized and the net AC voltage across the clear pixel should be minimized. The net DC voltage across any pixel should be zero.

The COM signal represents the time slice for each common, while the SEG contains the pixel data.

The pixel signal (COM-SEG) will have no DC component and can take only one of the two rms values. The higher rms value will create a dark pixel and a lower rms value will create a clear pixel.

As the number of commons increases, the delta between the two rms values decreases. The delta represents the maximum contrast that the display can have.

The LCDs can be driven by two types of waveforms: Type-A and Type-B. In a Type-A waveform, the phase changes within each common type, whereas a Type-B waveform's phase changes on each frame boundary. Thus, Type-A waveforms maintain 0 VDC over a single frame, whereas Type-B waveforms take two frames.

Note 1: If Sleep has to be executed with LCD Sleep enabled (SLPEN (LCDCON<6>) = 1), care must be taken to execute Sleep only when VDC on all the pixels is '0'.

2: When the LCD clock source is (FOSC/4)/8192, if Sleep is executed irrespective of the LCDCON<SLPEN> setting, the LCD goes into Sleep. Thus, take care to see that VDC on all pixels is '0' when Sleep is executed.

Figure 20-7 through Figure 20-17 provide waveforms for static, half-multiplex, one-third multiplex and quarter multiplex drives for Type-A and Type-B waveforms.

21.4.3.5 Reception

When the R/$\overline{W}$ bit of the address byte is clear and an address match occurs, the R/$\overline{W}$ bit of the SSPxSTAT register is cleared. The received address is loaded into the SSPxBUF register and the SDAx line is held low ($\overline{ACK}$).

When the address byte overflow condition exists, then the no Acknowledge ($\overline{ACK}$) pulse is given. An overflow condition is defined as either bit, BF (SSPxSTAT<0>), is set or bit, SSPOV (SSPxCON1<6>), is set.

An MSSP interrupt is generated for each data transfer byte. The interrupt flag bit, SSPxIF, must be cleared in software. The SSPxSTAT register is used to determine the status of the byte.

If SEN is enabled (SSPxCON2<0> = 1), SCLx will be held low (clock stretch) following each data transfer. The clock must be released by setting bit, CKP (SSPxCON1<4>). See **Section 21.4.4 "Clock Stretching"** for more details.

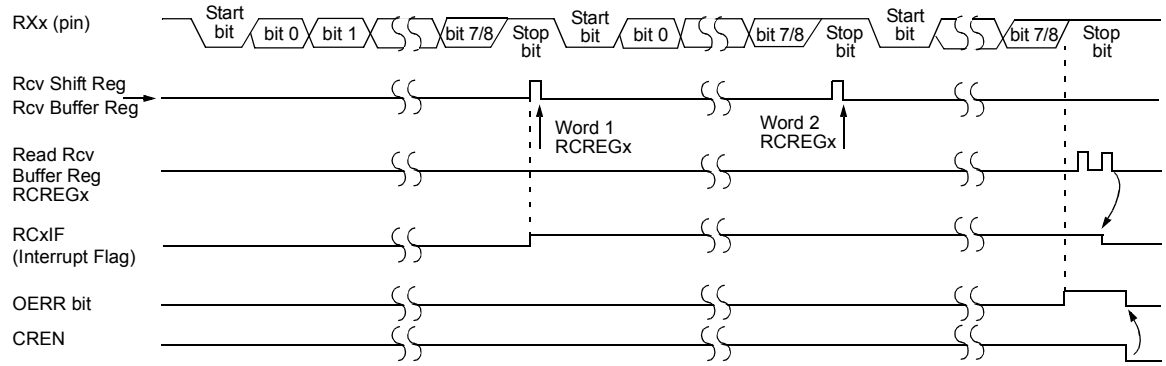
21.4.3.6 Transmission

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPxSTAT register is set. The received address is loaded into the SSPxBUF register. The $\overline{ACK}$ pulse will be sent on the ninth bit and pin, SCLx, is held low regardless of SEN (see **Section 21.4.4 "Clock Stretching"** for more details). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data. The transmit data must be loaded into the SSPxBUF register which also loads the SSPxSR register. Then, pin, SCLx, should be enabled by setting bit, CKP (SSPxCON1<4>). The eight data bits are shifted out on the falling edge of the SCLx input. This ensures that the SDAx signal is valid during the SCLx high time (Figure 21-10).

The $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCLx input pulse. If the SDAx line is high (not $\overline{ACK}$), then the data transfer is complete. In this case, when the $\overline{ACK}$ is latched by the slave, the slave logic is reset and the slave monitors for another occurrence of the Start bit. If the SDAx line was low ($\overline{ACK}$), the next transmit data must be loaded into the SSPxBUF register. Again, pin, SCLx, must be enabled by setting bit, CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPxIF bit must be cleared in software and the SSPxSTAT register is used to determine the status of the byte. The SSPxIF bit is set on the falling edge of the ninth clock pulse.

FIGURE 22-7: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows three words appearing on the RXx input. The RCREGx (Receive Buffer) is read after the third word causing the OERR (Overrun) bit to be set.

TABLE 22-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	75
PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	TMR1GIF	TMR2IF	TMR1IF	77
PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	TMR1GIE	TMR2IE	TMR1IE	77
IPR1	—	ADIP	RC1IP	TX1IP	SSP1IP	TMR1GIP	TMR2IP	TMR1IP	77
PIR3	TMR5GIF	LCDIF	RC2IF	TX2IF	CTMUIF	CCP2IF	CCP1IF	RTCCIF	77
PIE3	TMR5GIE	LCDIE	RC2IE	TX2IE	CTMUIE	CCP2IE	CCP1IE	RTCCIE	77
IPR3	TMR5GIP	LCDIP	RC2IP	TX2IP	CTMUIP	CCP2IP	CCP1IP	RTCCIP	77
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	77
RCREG1	EUSART1 Receive Register								77
TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	77
BAUDCON1	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	79
SPBRGH1	EUSART1 Baud Rate Generator Register High Byte								76
SPBRG1	EUSART1 Baud Rate Generator Register Low Byte								77
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	81
RCREG2	EUSART2 Receive Register								82
TXSTA2	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	81
BAUDCON2	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	81
SPBRGH2	EUSART2 Baud Rate Generator Register High Byte								82
SPBRG2	EUSART2 Baud Rate Generator Register Low Byte								82

Legend: — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

REGISTER 28-7: CONFIG4L: CONFIGURATION REGISTER 4 LOW (BYTE ADDRESS 300006h)

R/P-1	U-0	U-0	R/P-0	U-0	R/P-0	U-0	R/P-1
$\overline{\text{DEBUG}}$	—	—	BBSIZ0	—	—	—	STVREN
bit 7							bit 0

Legend:	P = Programmable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **$\overline{\text{DEBUG}}$:** Background Debugger Enable bit
1 = Background debugger is disabled, RB6 and RB7 are configured as general purpose I/O pins
0 = Background debugger is enabled, RB6 and RB7 are dedicated to In-Circuit Debug

bit 6-5 **Unimplemented:** Read as '0'

bit 4 **BBSIZ<0>:** Boot Block Size Select bit
1 = 2 kW boot block size
0 = 1 kW boot block size

bit 3-1 **Unimplemented:** Read as '0'

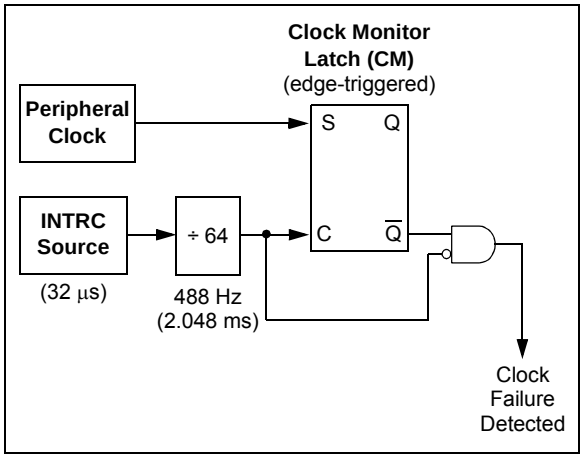
bit 0 **STVREN:** Stack Full/Underflow Reset Enable bit
1 = Stack full/underflow will cause a Reset
0 = Stack full/underflow will not cause a Reset

28.5 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the microcontroller to continue operation in the event of an external oscillator failure by automatically switching the device clock to the internal oscillator block. The FSCM function is enabled by setting the FCMEN Configuration bit.

When FSCM is enabled, the LF-INTOSC oscillator runs at all times to monitor clocks to peripherals and provide a backup clock in the event of a clock failure. Clock monitoring (shown in Figure 28-4) is accomplished by creating a sample clock signal, which is the output from the LF-INTOSC, divided by 64. This allows ample time between FSCM sample clocks for a peripheral clock edge to occur. The peripheral device clock and the sample clock are presented as inputs to the Clock Monitor (CM) latch. The CM is set on the falling edge of the device clock source, but cleared on the rising edge of the sample clock.

FIGURE 28-4: FSCM BLOCK DIAGRAM



Clock failure is tested for on the falling edge of the sample clock. If a sample clock falling edge occurs while CM is still set, a clock failure has been detected (Figure 28-5). This causes the following:

- The FSCM generates an oscillator fail interrupt by setting bit, OSCFIF (PIR2<7>)
- The device clock source switches to the internal oscillator block (OSCCON is not updated to show the current clock source – this is the Fail-Safe condition)
- The WDT is reset

During switchover, the postscaler frequency from the internal oscillator block may not be sufficiently stable for timing-sensitive applications. In these cases, it may be desirable to select another clock configuration and enter an alternate power-managed mode. This can be done to attempt a partial recovery or execute a controlled shutdown. See **Section 4.1.4 "Multiple Sleep Commands"** and **Section 28.4.1 "Special Considerations for Using Two-Speed Start-up"** for more details.

To use a higher clock speed on wake-up, the INTOSC or postscaler clock sources can be selected to provide a higher clock speed by setting bits, IRCF<2:0>, immediately after Reset. For wake-ups from Sleep, the INTOSC or postscaler clock sources can be selected by setting the IRCF<2:0> bits prior to entering Sleep mode.

The FSCM will detect only failures of the primary or secondary clock sources. If the internal oscillator block fails, no failure would be detected nor would any action be possible.

28.5.1 FSCM AND THE WATCHDOG TIMER

Both the FSCM and the WDT are clocked by the INTOSC oscillator. Since the WDT operates with a separate divider and counter, disabling the WDT has no effect on the operation of the INTOSC oscillator when the FSCM is enabled.

As already noted, the clock source is switched to the INTOSC clock when a clock failure is detected. Depending on the frequency selected by the IRCF<2:0> bits, this may mean a substantial change in the speed of code execution. If the WDT is enabled with a small prescale value, a decrease in clock speed allows a WDT time-out to occur and a subsequent device Reset. For this reason, Fail-Safe Clock events also reset the WDT and postscaler, allowing it to start timing from when execution speed was changed, and decreasing the likelihood of an erroneous time-out.

28.5.2 EXITING FAIL-SAFE OPERATION

The Fail-Safe condition is terminated by either a device Reset or by entering a power-managed mode. On Reset, the controller starts the primary clock source specified in Configuration Register 1H (with any required start-up delays that are required for the oscillator mode, such as the OST or PLL timer). The INTOSC multiplexer provides the device clock until the primary clock source becomes ready (similar to a Two-Speed Start-up). The clock source is then switched to the primary clock (indicated by the OSTS bit in the OSCCON register becoming set). The Fail-Safe Clock Monitor then resumes monitoring the peripheral clock.

The primary clock source may never become ready during start-up. In this case, operation is clocked by the INTOSC multiplexer. The OSCCON register will remain in its Reset state until a power-managed mode is entered.

FIGURE 31-8: HIGH/LOW-VOLTAGE DETECT CHARACTERISTICS

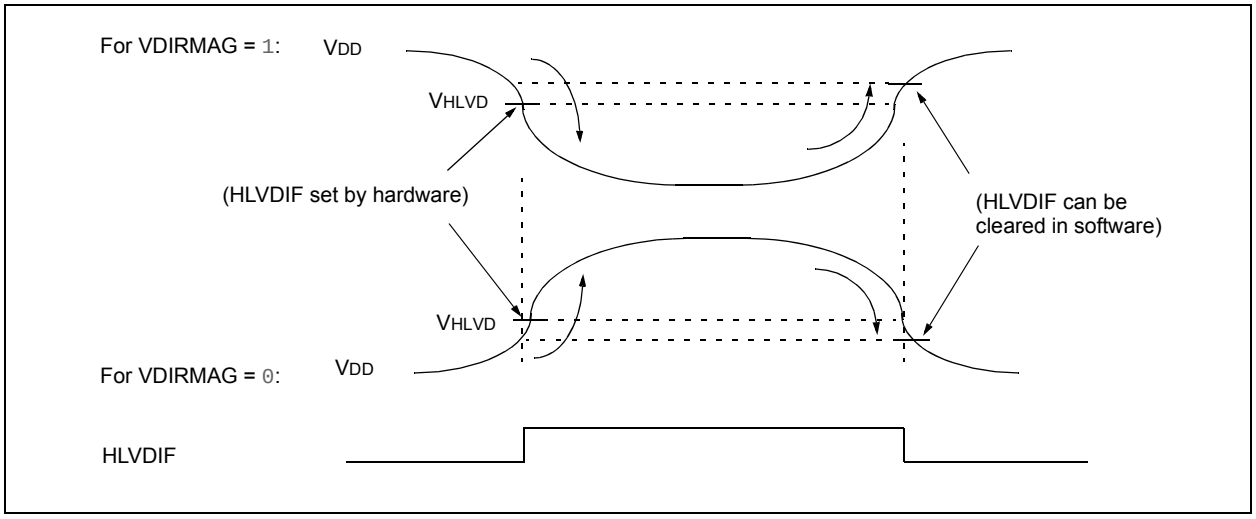


TABLE 31-11: HIGH/LOW-VOLTAGE DETECT CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended								
Param No.	**Sym**	**Characteristic**		**Min**	**Typ**	**Max**	**Units**	**Conditions**
D420		HLVD Voltage on VDD Transition High-to-Low	HLVDL<3:0> = 0000	1.80	1.86	1.90	V	
			HLVDL<3:0> = 0001	2.03	2.12	2.13	V	
			HLVDL<3:0> = 0010	2.24	2.33	2.35	V	
			HLVDL<3:0> = 0011	2.40	2.49	2.53	V	
			HLVDL<3:0> = 0100	2.50	2.59	2.62	V	
			HLVDL<3:0> = 0101	2.70	2.75	2.84	V	
			HLVDL<3:0> = 0110	2.82	2.93	2.97	V	
			HLVDL<3:0> = 0111	2.95	3.07	3.10	V	
			HLVDL<3:0> = 1000	3.24	3.30	3.41	V	
			HLVDL<3:0> = 1001	3.42	3.48	3.59	V	
			HLVDL<3:0> = 1010	3.61	3.67	3.79	V	
			HLVDL<3:0> = 1011	3.82	3.87	4.01	V	
			HLVDL<3:0> = 1100	4.06	4.21	4.26	V	
			HLVDL<3:0> = 1101	4.33	4.42	4.55	V	
			HLVDL<3:0> = 1110	4.64	4.77	4.87	V	

NOTES: