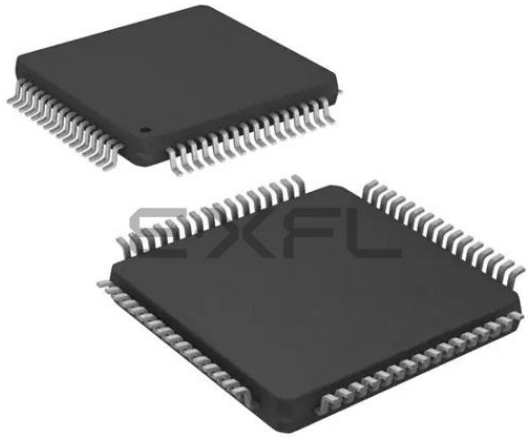
Microchip Technology - DSPIC33EV64GM106-E/PT Datasheet



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	dsPIC
Core Size	16-Bit
Speed	60 MIPs
Connectivity	CANbus, I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, Motor Control PWM, POR, PWM, WDT
Number of I/O	53
Program Memory Size	64KB (22K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 36x10/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33ev64gm106-e-pt

TABLE 4-16: PERIPHERAL PIN SELECT OUTPUT REGISTER MAP FOR dsPIC33EVXXXGM006/106 DEVICES

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
RPOR0	0670	—	—	RP35R5	RP35R4	RP35R3	RP35R2	RP35R1	RP35R0	—	—	RP20R5	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0	0000
RPOR1	0672	—	—	RP37R5	RP37R4	RP37R3	RP37R2	RP37R1	RP37R0	—	—	RP36R5	RP36R4	RP36R3	RP36R2	RP36R1	RP36R0	0000
RPOR2	0674	—	—	RP39R5	RP39R4	RP39R3	RP39R2	RP39R1	RP39R0	—	—	RP38R5	RP38R4	RP38R3	RP38R2	RP38R1	RP38R0	0000
RPOR3	0676	—	—	RP41R5	RP41R4	RP41R3	RP41R2	RP41R1	RP41R0	—	—	RP40R5	RP40R4	RP40R3	RP40R2	RP40R1	RP40R0	0000
RPOR4	0678	—	—	RP43R5	RP43R4	RP43R3	RP43R2	RP43R1	RP43R0	—	—	RP42R5	RP42R4	RP42R3	RP42R2	RP42R1	RP42R0	0000
RPOR5	067A	—	—	RP49R5	RP49R4	RP49R3	RP49R2	RP49R1	RP49R0	—	—	RP48R5	RP48R4	RP48R3	RP48R2	RP48R1	RP48R0	0000
RPOR6	067C	—	—	RP55R5	RP55R4	RP55R3	RP55R2	RP55R1	RP55R0	—	—	RP54R5	RP54R4	RP54R3	RP54R2	RP54R1	RP54R0	0000
RPOR7	067E	—	—	RP57R5	RP57R4	RP57R3	RP57R2	RP57R1	RP57R0	—	—	RP56R5	RP56R4	RP56R3	RP56R2	RP56R1	RP56R0	0000
RPOR8	0680	—	—	RP70R5	RP70R4	RP70R3	RP70R2	RP70R1	RP70R0	—	—	RP69R5	RP69R4	RP69R3	RP69R2	RP69R1	RP69R0	0000
RPOR9	0682	—	—	RP118R5	RP118R4	RP118R3	RP118R2	RP118R1	RP118R0	—	—	RP97R5	RP97R4	RP97R3	RP97R2	RP97R1	RP97R0	0000
RPOR10	0684	—	—	RP176R5	RP176R4	RP176R3	RP176R2	RP176R1	RP176R0	—	—	RP120R5	RP120R4	RP120R3	RP120R2	RP120R1	RP120R0	0000
RPOR11	0686	—	—	RP178R5	RP178R4	RP178R3	RP178R2	RP178R1	RP178R0	—	—	RP177R5	RP177R4	RP177R3	RP177R2	RP177R1	RP177R0	0000
RPOR12	0688	—	—	RP180R5	RP180R4	RP180R3	RP180R2	RP180R1	RP180R0	—	—	RP179R5	RP179R4	RP179R3	RP179R2	RP179R1	RP179R0	0000
RPOR13	068A	—	—	—	—	—	—	—	—	—	—	RP181R<5:0>						0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-23: INTERRUPT CONTROLLER REGISTER MAP FOR dsPIC33EVXXXGM00X/10X FAMILY DEVICES

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
IFS0	0800	NVMIF	DMA1IF	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPI1EIF	T3IF	T2IF	OC2IF	IC2IF	DMA0IF	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0802	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	DMA2IF	—	—	—	INT1IF	CNIF	CMPIF	MI2C1IF	SI2C1IF	0000
IFS2	0804	—	—	—	—	—	—	—	—	—	IC4IF	IC3IF	DMA3IF	C1IF	C1RXIF(1)	SPI2IF	SPI2EIF	0000
IFS3	0806	—	—	—	—	—	—	PSEMIF	—	—	—	—	—	—	—	—	—	0000
IFS4	0808	—	—	CTMUIF	—	—	—	—	—	—	C1TXIF(1)	—	—	—	U2EIF	U1EIF	—	0000
IFS5	080A	PWM2IF	PWM1IF	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IFS6	080C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWM3IF	0000
IFS8	0810	—	ICDIF	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IFS10	0814	—	—	I2C1BCIF	—		—	—	—	—	—	—	—	—	—	—	—	0000
IFS11	0816	—	—	—	—	—	ECCSBEIF	SENT2IF	SENT2EIF	SENT1IF	SENT1EIF	—	—	—	—	—	—	0000
IEC0	0820	NVMIE	DMA1IE	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPI1EIE	T3IE	T2IE	OC2IE	IC2IE	DMA0IE	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0822	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	DMA2IE	—	—	—	INT1IE	CNIE	CMPIE	MI2C1IE	SI2C1IE	0000
IEC2	0824	—	—	—	—	—	—	—	—	—	IC4IE	IC3IE	DMA3IE	C1IE	C1RXIE(1)	SPI2IE	SPI2EIE	0000
IEC3	0826	—	—	—	—	—	—	PSEMIE	—	—	—	—	—	—	—	—	—	0000
IEC4	0828	—	—	CTMUIE	—	—	—	—	—	—	C1TXIE(1)	—	—	—	U2EIE	U1EIE	—	0000
IEC5	082A	PWM2IE	PWM1IE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IEC6	082C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWM3IE	0000
IEC8	0830	—	ICDIE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IEC10	0834	—	—	I2C1BCIE	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IEC11	0836	—	—	—	—	—	ECCSBEIE	SENT2IE	SENT2EIE	SENT1IE	SENT1EIE	—	—	—	—	—	—	0000
IPC0	0840	—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0	—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0	4444
IPC1	0842	—	T2IP2	T2IP1	T2IP0	—	OC2IP2	OC2IP1	OC2IP0	—	IC2IP2	IC2IP1	IC2IP0	—	DMA0IP2	DMA0IP1	DMA0IP0	4444
IPC2	0844	—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0	—	SPI1EIP2	SPI1EIP1	SPI1EIP0	—	T3IP2	T3IP1	T3IP0	4444
IPC3	0846	—	NVMIP2	NVMIP1	NVMIP0	—	DMA1IP2	DMA1IP1	DMA1IP0	—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0	4444
IPC4	0848	—	CNIP2	CNIP1	CNIP0	—	CMPIP2	CMPIP1	CMPIP0	—	MI2C1IP2	MI2C1IP1	MI2C1IP0	—	SI2C1IP2	SI2C1IP1	SI2C1IP0	4444
IPC5	084A	—	—	—	—	—	—	—	—	—	—	—	—	—		INT1IP<2:0>		0004
IPC6	084C	—	T4IP2	T4IP1	T4IP0	—	OC4IP2	OC4IP1	OC4IP0	—	OC3IP2	OC3IP1	OC3IP0	—	DMA2IP2	DMA2IP1	DMA2IP0	4444
IPC7	084E	—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0	—	INT2IP2	INT2IP1	INT2IP0	—	T5IP2	T5IP1	T5IP0	4444
IPC8	0850	—	C1IP2	C1IP1	C1IP0	—	C1RXIP2(1)	C1RXIP1(1)	C1RXIP0(1)	—	SPI2IP2	SPI2IP1	SPI2IP0	—	SPI2EIP2	SPI2EIP1	SPI2EIP0	4444
IPC9	0852	—	—	—	—	—	IC4IP2	IC4IP1	IC4IP0	—	IC3IP2	IC3IP1	IC3IP0	—	DMA3IP2	DMA3IP1	DMA3IP0	0444
IPC14	085C	—	—	—	—	—	—	—	—	—	PSEMIP<2:0>			—	—	—	—	0040
IPC16	0860	—	—	—	—	—	U2EIP2	U2EIP1	U2EIP0	—	U1EIP2	U1EIP1	U1EIP0	—	—	—	—	0440
IPC17	0862	—	—	—	—	—	C1TXIP<2:0>(1)			—	—	—	—	—	—	—	—	0400

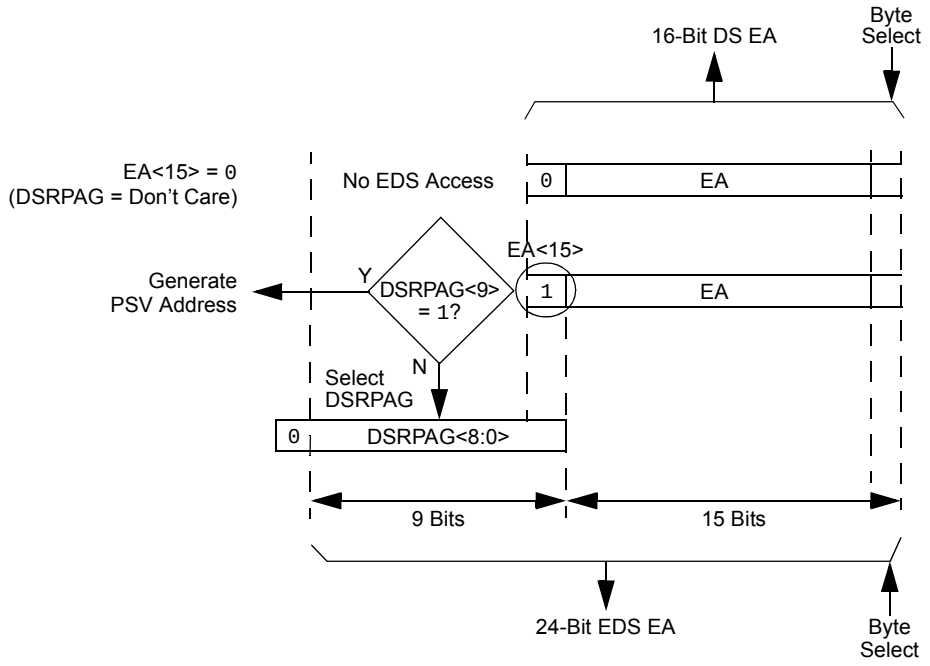
Legend: — = unimplemented, read as '0' Reset values are shown in hexadecimal.

Note 1: This feature is available only on dsPIC33EVXXXGM10X devices.

4.3.1 PAGED MEMORY SCHEME

The dsPIC33EVXXXGM00X/10X family architecture extends the available DS through a paging scheme, which allows the available DS to be accessed using MOV instructions in a linear fashion for pre- and post-modified Effective Addresses (EAs). The upper half of the Base Data Space address is used in conjunction with the Data Space Page registers, the 10-bit Data Space Read Page register (DSRPAG) or the 9-bit Data Space Write Page register (DSWPAG), to form an EDS address, or Program Space Visibility (PSV) address. The Data Space Page registers are located in the SFR space. Construction of the EDS address is shown in Figure 4-9 and Figure 4-10. When DSRPAG<9> = 0 and the base address bit, EA<15> = 1, the DSRPAG<8:0> bits are concatenated onto EA<14:0> to form the 24-bit EDS read address. Similarly, when the base address bit, EA<15> = 1, the DSWPAG<8:0> bits are concatenated onto EA<14:0> to form the 24-bit EDS write address.

FIGURE 4-9: EXTENDED DATA SPACE (EDS) READ ADDRESS GENERATION



Note: DS read access when DSRPAG = 0x000 will force an address error trap.

REGISTER 7-6: INTCON4: INTERRUPT CONTROL REGISTER 4

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R-0, HS, SC	R-0, HS, SC
—	—	—	—	—	—	ECCDBE(1)	SGHT
bit 7							bit 0

Legend:	HS = Hardware Settable bit	SC = Software Clearable 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 15-2 **Unimplemented:** Read as '0'

bit 1 **ECCDBE:** ECC Double-Bit Error Trap bit(1)
1 = ECC double-bit error trap has occurred
0 = ECC double-bit error trap has not occurred

bit 0 **SGHT:** Software-Generated Hard Trap Status bit
1 = Software-generated hard trap has occurred
0 = Software-generated hard trap has not occurred

Note 1: ECC double-bit error causes a generic hard trap.

REGISTER 11-16: RPINR44: PERIPHERAL PIN SELECT INPUT REGISTER 44

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SENT1R<7:0>							
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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 15-8 **SENT1R<7:0>:** Assign SENT Module Input 1 to the Corresponding RPn Pin bits
(see Table 11-2 for input pin selection numbers)
10110101 = Input tied to RPI181
•
•
•
00000001 = Input tied to CMP1
00000000 = Input tied to Vss

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

REGISTER 11-17: RPINR45: PERIPHERAL PIN SELECT INPUT REGISTER 45

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SENT2R<7:0>							
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 15-8 **Unimplemented:** Read as '0'

bit 7-0 **SENT2R<7:0>:** Assign SENT Module Input 2 to the Corresponding RPn Pin bits
(see Table 11-2 for input pin selection numbers)
10110101 = Input tied to RPI181
•
•
•
00000001 = Input tied to CMP1
00000000= Input tied to Vss

REGISTER 11-26: RPOR8: PERIPHERAL PIN SELECT OUTPUT REGISTER 8(1)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP70R5	RP70R4	RP70R3	RP70R2	RP70R1	RP70R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP69R5	RP69R4	RP69R3	RP69R2	RP69R1	RP69R0
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 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP70R<5:0>:** Peripheral Output Function is Assigned to RP70 Output Pin bits (see Table 11-3 for peripheral function numbers)

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

bit 5-0 **RP69R<5:0>:** Peripheral Output Function is Assigned to RP69 Output Pin bits (see Table 11-3 for peripheral function numbers)

Note 1: This register is present in dsPIC33EVXXXGM004/104/006/106 devices only.

REGISTER 11-27: RPOR9: PERIPHERAL PIN SELECT OUTPUT REGISTER 9(1)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP118R5	RP118R4	RP118R3	RP118R2	RP118R1	RP118R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP97R5	RP97R4	RP97R3	RP97R2	RP97R1	RP97R0
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 15-14 **Unimplemented:** Read as '0'

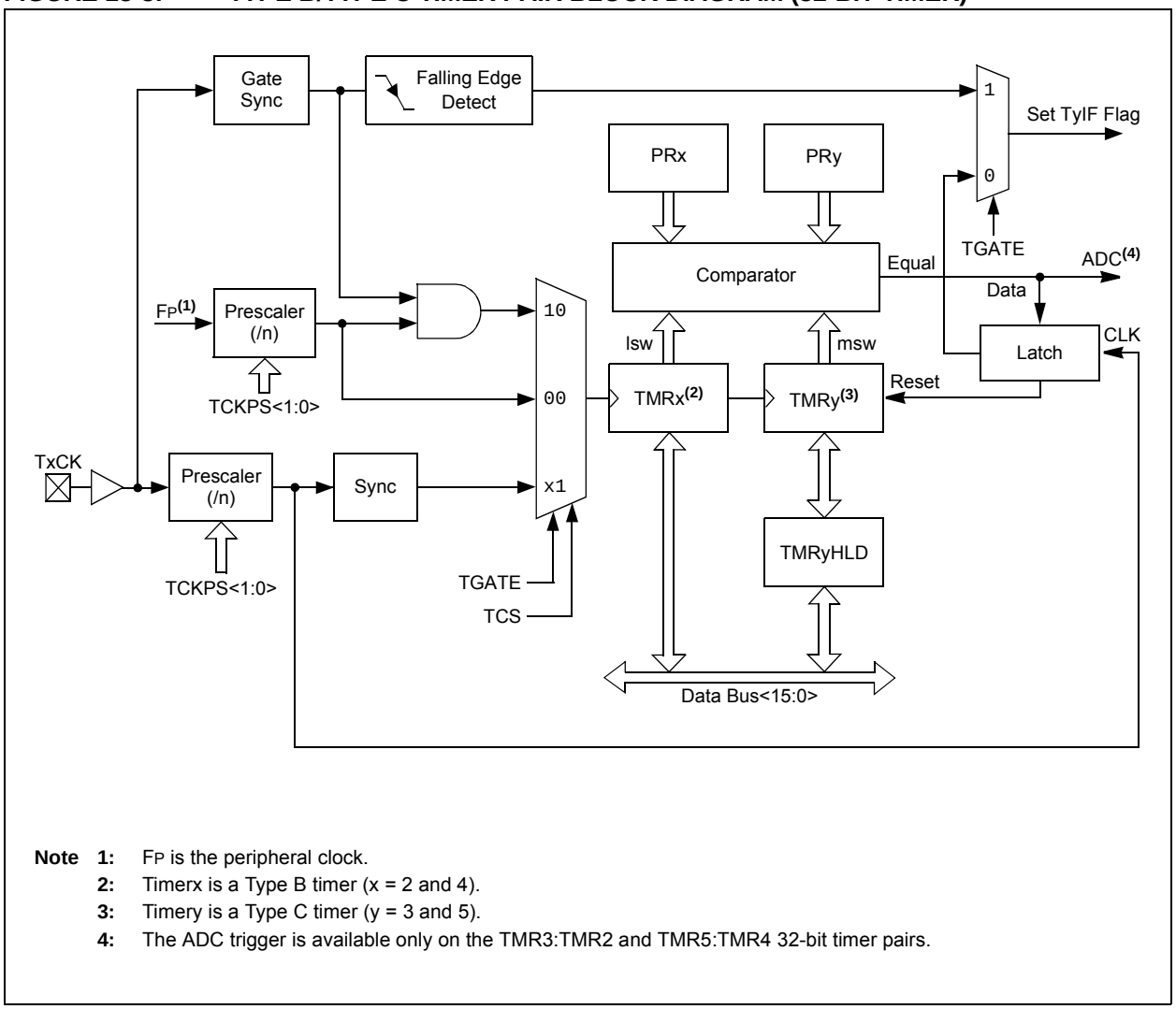
bit 13-8 **RP118R<5:0>:** Peripheral Output Function is Assigned to RP118 Output Pin bits (see Table 11-3 for peripheral function numbers)

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

bit 5-0 **RP97R<5:0>:** Peripheral Output Function is Assigned to RP97 Output Pin bits (see Table 11-3 for peripheral function numbers)

Note 1: This register is present in dsPIC33EVXXXGM004/106 devices only.

FIGURE 13-3: TYPE B/TYPE C TIMER PAIR BLOCK DIAGRAM (32-BIT TIMER)



Note 1: FP is the peripheral clock.
2: Timerx is a Type B timer (x = 2 and 4).
3: Timery is a Type C timer (y = 3 and 5).
4: The ADC trigger is available only on the TMR3:TMR2 and TMR5:TMR4 32-bit timer pairs.

REGISTER 13-2: TyCON (T3CON AND T5CON) CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON(1)	—	TSIDL(2)	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
—	TGATE(1)	TCKPS1(1)	TCKPS0(1)	—	—	TCS(1,3)	—
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 15 **TON:** Timery On bit(1)
1 = Starts 16-bit Timery
0 = Stops 16-bit Timery

bit 14 **Unimplemented:** Read as '0'

bit 13 **TSIDL:** Timery Stop in Idle Mode bit(2)
1 = Discontinues module operation when the device enters an Idle mode
0 = Continues module operation in an Idle mode

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **TGATE:** Timery Gated Time Accumulation Enable bit(1)
When TCS = 1:
This bit is ignored.
When TCS = 0:
1 = Gated time accumulation is enabled
0 = Gated time accumulation is disabled

bit 5-4 **TCKPS<1:0>:** Timery Input Clock Prescale Select bits(1)
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **TCS:** Timery Clock Source Select bit(1,3)
1 = External clock is from pin, TyCK (on the rising edge)
0 = Internal clock (FP)

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

Note 1: When 32-bit operation is enabled (T2CON<3> = 1), these bits have no effect on Timery operation; all timer functions are set through TxCON.

2: When 32-bit timer operation is enabled (T32 = 1) in the Timerx Control register (TxCON<3>), the TSIDL bit must be cleared to operate the 32-bit timer in Idle mode.

3: The TyCK pin is not available on all timers. See the **"Pin Diagrams"** section for the available pins.

18.1 SPI Helpful Tips

1. In Frame mode, if there is a possibility that the master may not be initialized before the slave:
   a) If FRMPOL (SPIxCON2<13>) = 1, use a pull-down resistor on $\overline{SSx}$.
   b) If FRMPOL = 0, use a pull-up resistor on $\overline{SSx}$.

Note:	This insures that the first frame transmission after initialization is not shifted or corrupted.

2. In Non-Framed 3-Wire mode (i.e., not using $\overline{SSx}$ from a master):
   a) If CKP (SPIxCON1<6>) = 1, always place a pull-up resistor on $\overline{SSx}$.
   b) If CKP = 0, always place a pull-down resistor on $\overline{SSx}$.

Note:	This will insure that during power-up and initialization, the master/slave will not lose sync due to an errant SCKx transition that would cause the slave to accumulate data shift errors, for both transmit and receive, appearing as corrupted data.

3. FRMEN (SPIxCON2<15>) = 1 and SSEN (SPIxCON1<7>) = 1 are exclusive and invalid. In Frame mode, SCKx is continuous and the Frame Sync pulse is active on the $\overline{SSx}$ pin, which indicates the start of a data frame.

Note:	Not all third-party devices support Frame mode timing. For more information, refer to the SPI specifications in **Section 30.0 "Electrical Characteristics"**.

4. In Master mode only, set the SMP bit (SPIxCON1<9>) to a '1' for the fastest SPI data rate possible. The SMP bit can only be set at the same time or after the MSTEN bit (SPIxCON1<5>) is set.

To avoid invalid slave read data to the master, the user's master software must ensure enough time for slave software to fill its write buffer before the user application initiates a master write/read cycle. It is always advisable to preload the SPIxBUF Transmit register in advance of the next master transaction cycle. SPIxBUF is transferred to the SPIx Shift register and is empty once the data transmission begins.

18.2 SPI Control Registers

REGISTER 18-1: SPIxSTAT: SPIx STATUS AND CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
SPIEN	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0
bit 15							bit 8

R/W-0	R/C-0, HS	R/W-0	R/W-0	R/W-0	R/W-0	R-0, HS, HC	R-0, HS, HC
SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF
bit 7							bit 0

Legend:	HC = Hardware Clearable bit	HS = Hardware Settable 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	C = Clearable bit

bit 15 **SPIEN:** SPIx Enable bit
1 = Enables the SPIx module and configures SCKx, SDOx, SDIx and $\overline{SSx}$ as serial port pins
0 = Disables the SPIx module

bit 14 **Unimplemented:** Read as '0'

bit 13 **SPISIDL:** SPIx Stop in Idle Mode bit
1 = Discontinues the SPIx module operation when the device enters Idle mode
0 = Continues the SPIx module operation in Idle mode

bit 12-11 **Unimplemented:** Read as '0'

bit 10-8 **SPIBEC<2:0>:** SPIx Buffer Element Count bits (valid in Enhanced Buffer mode)
Master mode:
Number of SPIx transfers are pending.
Slave mode:
Number of SPIx transfers are unread.

bit 7 **SRMPT:** SPIx Shift Register (SPIxSR) Empty bit (valid in Enhanced Buffer mode)
1 = The SPIx Shift register is empty and ready to send or receive the data
0 = The SPIx Shift register is not empty

bit 6 **SPIROV:** SPIx Receive Overflow Flag bit
1 = A new byte/word is completely received and discarded; the user application has not read the previous data in the SPIxBUF register
0 = Overflow has not occurred

bit 5 **SRXMPT:** SPIx Receive FIFO Empty bit (valid in Enhanced Buffer mode)
1 = RX FIFO is empty
0 = RX FIFO is not empty

bit 4-2 **SISEL<2:0>:** SPIx Buffer Interrupt Mode bits (valid in Enhanced Buffer mode)
111 = Interrupt when the SPIx transmit buffer is full (SPITBF bit is set)
110 = Interrupt when the last bit is shifted into SPIxSR, and as a result, the TX FIFO is empty
101 = Interrupt when the last bit is shifted out of SPIxSR and the transmit is complete
100 = Interrupt when one data is shifted into SPIxSR, and as a result, the TX FIFO has one open memory location
011 = Interrupt when the SPIx receive buffer is full (SPIRBF bit is set)
010 = Interrupt when the SPIx receive buffer is 3/4 or more full
001 = Interrupt when data is available in the SPIx receive buffer (SRMPT bit is set)
000 = Interrupt when the last data in the SPIx receive buffer is read, and as a result, the buffer is empty (SRXMPT bit is set)

21.0 UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

Note 1: This data sheet summarizes the features of the dsPIC33EVXXXGM00X/10X family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Universal Asynchronous Receiver Transmitter (UART)"** (DS70000582) in the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

2: Some registers and associated bits described in this section may not be available on all devices. Refer to **Section 4.0 "Memory Organization"** in this data sheet for device-specific register and bit information.

The dsPIC33EVXXXGM00X/10X family of devices contains two UART modules.

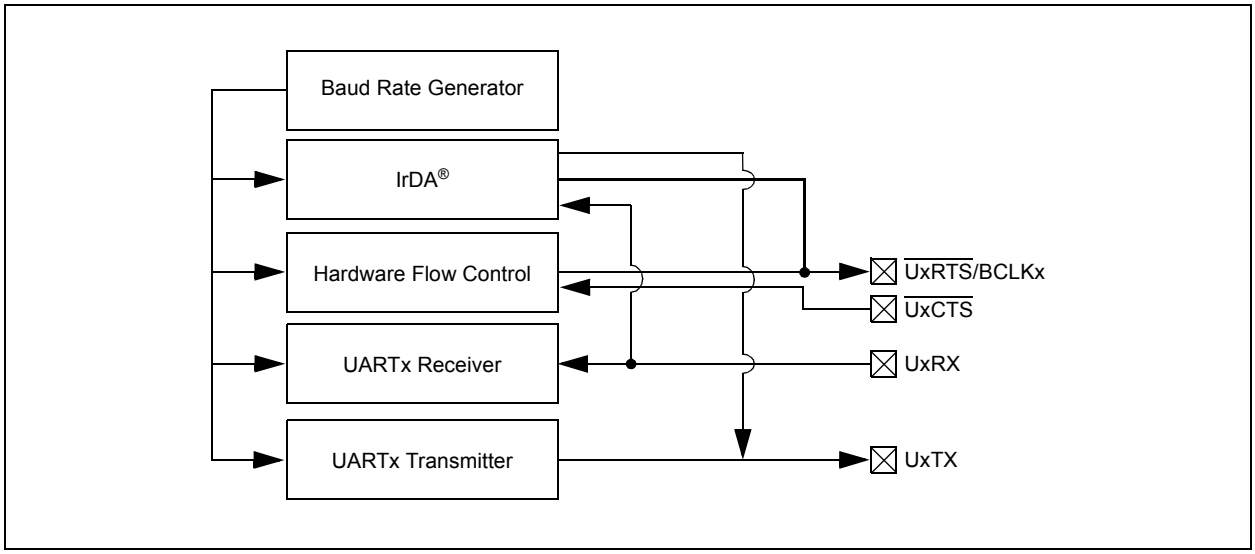
The Universal Asynchronous Receiver Transmitter (UART) module is one of the serial I/O modules available in the dsPIC33EVXXXGM00X/10X device family. The UART is a full-duplex, asynchronous system that can communicate with peripheral devices, such as personal computers, LIN/J2602, RS-232 and RS-485 interfaces. The module also supports a hardware flow control option with the $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ pins, and also includes an IrDA® encoder and decoder.

Note: Hardware flow control using $\overline{\text{UxRTS}}$ and $\overline{\text{UxCTS}}$ is not available on all pin count devices. See the **"Pin Diagrams"** section for availability.

The primary features of the UARTx module are:

- Full-Duplex, 8 or 9-Bit Data Transmission through the UxTX and UxRX Pins
- Even, Odd or No Parity Options (for 8-bit data)
- One or Two Stop Bits
- Hardware Flow Control Option with $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ Pins
- Fully Integrated Baud Rate Generator with 16-Bit Prescaler
- Baud Rates Ranging from 4.375 Mbps to 67 bps at 16x mode at 70 MIPS
- Baud Rates Ranging from 17.5 Mbps to 267 bps at 4x mode at 70 MIPS
- 4-Deep First-In First-Out (FIFO) Transmit Data Buffer
- 4-Deep FIFO Receive Data Buffer
- Parity, Framing and Buffer Overrun Error Detection
- Support for 9-Bit mode with Address Detect (9^{th} bit = 1)
- Transmit and Receive Interrupts
- A Separate Interrupt for All UART Error Conditions

FIGURE 21-1: UARTx SIMPLIFIED BLOCK DIAGRAM



22.0 CONTROLLER AREA NETWORK (CAN) MODULE (dsPIC33EVXXXGM10X DEVICES ONLY)

Note 1: This data sheet summarizes the features of the dsPIC33EVXXXGM00X/10X family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Enhanced Controller Area Network (ECAN™)"** (DS70353) in the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

2: Some registers and associated bits described in this section may not be available on all devices. Refer to **Section 4.0 "Memory Organization"** in this data sheet for device-specific register and bit information.

22.1 Overview

The Controller Area Network (CAN) module is a serial interface, useful for communicating with other CAN modules or microcontroller devices. This interface/ protocol was designed to allow communications within noisy environments. The dsPIC33EVXXXGM10X devices contain one CAN module.

The CAN module is a communication controller implementing the CAN 2.0 A/B protocol, as defined in the BOSCH CAN specification. The module supports CAN 1.2, CAN 2.0A, CAN 2.0B Passive and CAN 2.0B Active versions of the protocol. The module implementation is a full CAN system. The CAN specification is not covered within this data sheet. The reader can refer to the BOSCH CAN specification for further details.

The CAN module features are as follows:

- Implementation of the CAN Protocol, CAN 1.2, CAN 2.0A and CAN 2.0B
- Standard and Extended Data Frames
- 0 to 8-Byte Data Length
- Programmable Bit Rate, up to 1 Mbit/sec
- Automatic Response to Remote Transmission Requests
- Up to Eight Transmit Buffers with Application Specified Prioritization and Abort Capability (each buffer can contain up to 8 bytes of data)
- Up to 32 Receive Buffers (each buffer can contain up to 8 bytes of data)
- Up to 16 Full (Standard/Extended Identifier) Acceptance Filters
- Three Full Acceptance Filter Masks
- DeviceNet™ Addressing Support
- Programmable Wake-up Functionality with Integrated Low-Pass Filter
- Programmable Loopback Mode Supports Self-Test Operation
- Signaling through Interrupt Capabilities for All CAN Receiver and Transmitter Error States
- Programmable Clock Source
- Programmable Link to Input Capture 2 (IC2) module for Timestamping and Network Synchronization
- Low-Power Sleep and Idle Modes

The CAN bus module consists of a protocol engine and message buffering/control. The CAN protocol engine handles all functions for receiving and transmitting messages on the CAN bus. Messages are transmitted by first loading the appropriate data registers. Status and errors can be checked by reading the appropriate registers. Any message detected on the CAN bus is checked for errors, and then matched against filters to see if it should be received and stored in one of the Receive registers.

Figure 22-1 shows a block diagram of the CANx module.

22.3 CAN Control Registers

REGISTER 22-1: CxCTRL1: CANx CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0
—	—	CSIDL	ABAT	CANCKS	REQOP2	REQOP1	REQOP0
bit 15							bit 8

R-1	R-0	R-0	U-0	R/W-0	U-0	U-0	R/W-0
OPMODE2	OPMODE1	OPMODE0	—	CANCAP	—	—	WIN
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 15-14 **Unimplemented:** Read as '0'

bit 13 **CSIDL:** CANx Stop in Idle Mode bit
1 = Discontinues module operation when the device enters Idle mode
0 = Continues module operation in Idle mode

bit 12 **ABAT:** Abort All Pending Transmissions bit
1 = Signals all transmit buffers to abort transmission
0 = Module will clear this bit when all transmissions are aborted

bit 11 **CANCKS:** CANx Module Clock (FCAN) Source Select bit
1 = FCAN is equal to 2 * FP
0 = FCAN is equal to FP

bit 10-8 **REQOP<2:0>:** Request Operation Mode bits
111 = Sets Listen All Messages mode
110 = Reserved
101 = Reserved
100 = Sets Configuration mode
011 = Sets Listen Only mode
010 = Sets Loopback mode
001 = Sets Disable mode
000 = Sets Normal Operation mode

bit 7-5 **OPMODE<2:0>**: Operation Mode bits
111 = Module is in Listen All Messages mode
110 = Reserved
101 = Reserved
100 = Module is in Configuration mode
011 = Module is in Listen Only mode
010 = Module is in Loopback mode
001 = Module is in Disable mode
000 = Module is in Normal Operation mode

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

bit 3 **CANCAP:** CANx Message Receive Timer Capture Event Enable bit
1 = Enables input capture based on CAN message receive
0 = Disables CAN capture

bit 2-1 **Unimplemented:** Read as '0'

bit 0 **WIN:** SFR Map Window Select bit
1 = Uses filter window
0 = Uses buffer window

REGISTER 24-1: ADxCON1: ADCx CONTROL REGISTER 1 (CONTINUED)

bit 7-5 **SSRC<2:0>:** Sample Clock Source Select bits

If SSRCG = 1:
111 = Reserved
110 = Reserved
101 = Reserved
100 = Reserved
011 = Reserved
010 = PWM Generator 3 primary trigger compare ends sampling and starts conversion
001 = PWM Generator 2 primary trigger compare ends sampling and starts conversion
000 = PWM Generator 1 primary trigger compare ends sampling and starts conversion

If SSRCG = 0:
111 = Internal counter ends sampling and starts conversion (auto-convert)
110 = CTMU ends sampling and starts conversion
101 = Reserved
100 = Timer5 compare ends sampling and starts conversion
011 = PWM primary Special Event Trigger ends sampling and starts conversion
010 = Timer3 compare ends sampling and starts conversion
001 = Active transition on the INT0 pin ends sampling and starts conversion
000 = Clearing the Sample bit (SAMP) ends sampling and starts conversion (Manual mode)

bit 4 **SSRCG:** Sample Trigger Source Group bit

See SSRC<2:0> for details.

bit 3 **SIMSAM:** Simultaneous Sample Select bit (only applicable when CHPS<1:0> = 01 or 1x)

In 12-Bit Mode (AD12B = 1), SIMSAM is Unimplemented and is Read as '0':
1 = Samples CH0, CH1, CH2, CH3 simultaneously (when CHPS<1:0> = 1x) or samples CH0 and CH1 simultaneously (when CHPS<1:0> = 01)
0 = Samples multiple channels individually in sequence

bit 2 **ASAM:** ADCx Sample Auto-Start bit

1 = Sampling begins immediately after last conversion; SAMP bit is auto-set
0 = Sampling begins when SAMP bit is set

bit 1 **SAMP:** ADCx Sample Enable bit

1 = ADCx Sample-and-Hold amplifiers are sampling
0 = ADCx Sample-and-Hold amplifiers are holding
If ASAM = 0, software can write '1' to begin sampling. Automatically set by hardware if ASAM = 1. If SSRC<2:0> = 000, software can write '0' to end sampling and start conversion. If SSRC<2:0> ≠ 000, automatically cleared by hardware to end sampling and start conversion.

bit 0 **DONE:** ADCx Conversion Status bit(1)

1 = ADCx conversion cycle is completed.
0 = ADCx conversion has not started or is in progress
Automatically set by hardware when conversion is complete. Software can write '0' to clear DONE bit status (software not allowed to write '1'). Clearing this bit does NOT affect any operation in progress. Automatically cleared by hardware at the start of a new conversion.

Note 1: Do not clear the DONE bit in software if auto-sample is enabled (ASAM = 1).

Figure 25-2, shows the user-programmable blanking function block diagram.

FIGURE 25-2: USER-PROGRAMMABLE BLANKING FUNCTION BLOCK DIAGRAM

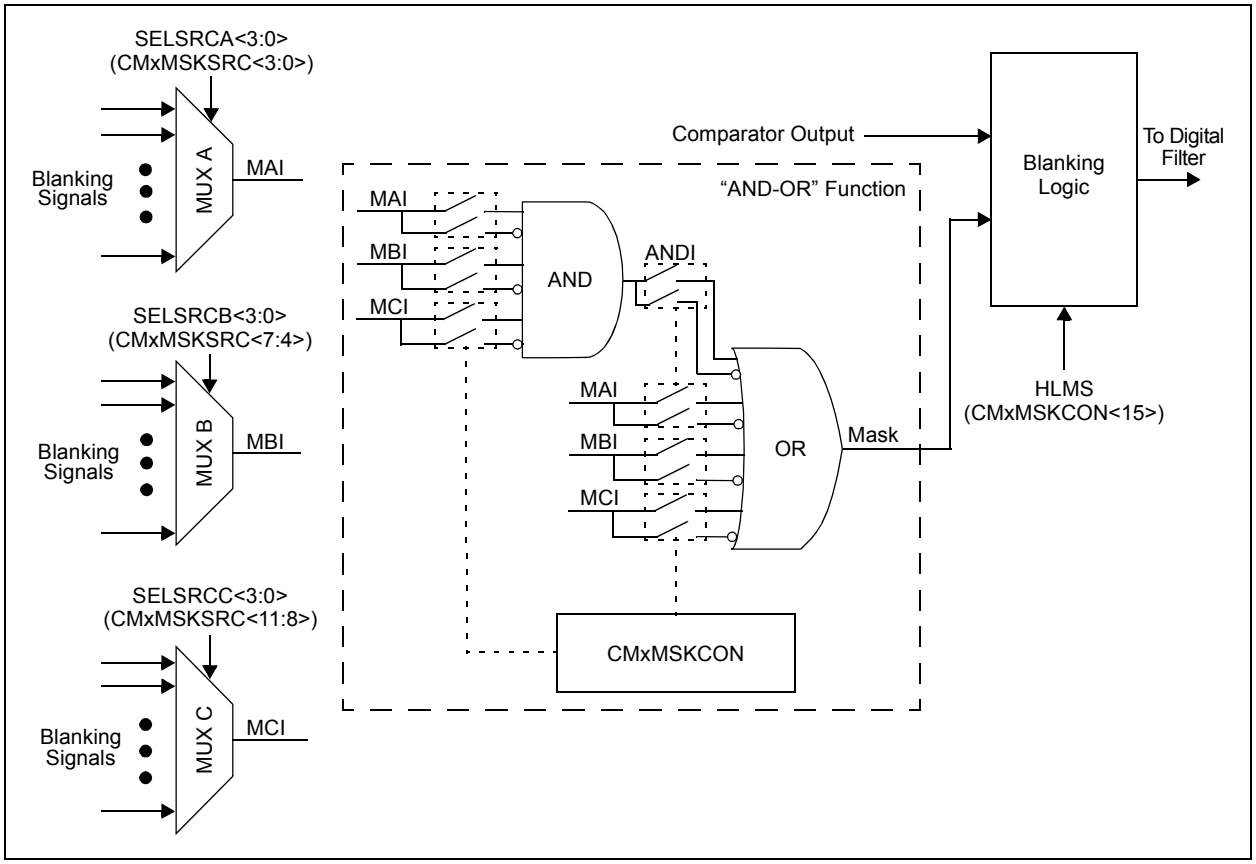
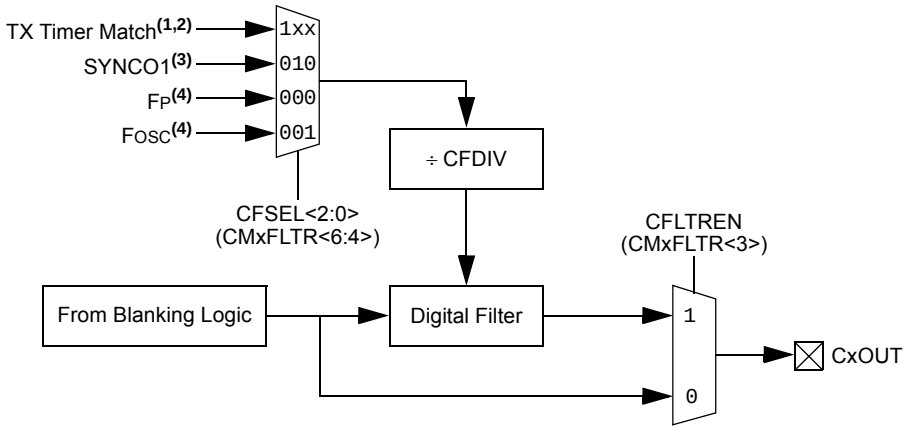


Figure 25-3, shows the digital filter interconnect block diagram.

FIGURE 25-3: DIGITAL FILTER INTERCONNECT BLOCK DIAGRAM



Note 1: See the Type C Timer Block Diagram (Figure 13-2).
2: See the Type B Timer Block Diagram (Figure 13-1).
3: See the PWMx Module Register Interconnect Diagram (Figure 17-2).
4: See the Oscillator System Diagram (Figure 9-1).

29.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

29.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

29.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

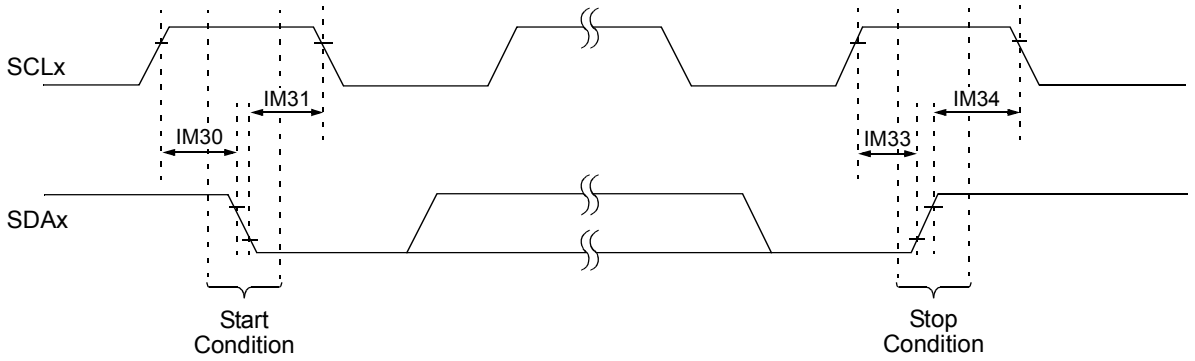
- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

29.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

FIGURE 30-28: I2Cx BUS START/STOP BITS TIMING CHARACTERISTICS (MASTER MODE)



Note: Refer to Figure 30-1 for load conditions.

FIGURE 30-29: I2Cx BUS DATA TIMING CHARACTERISTICS (MASTER MODE)

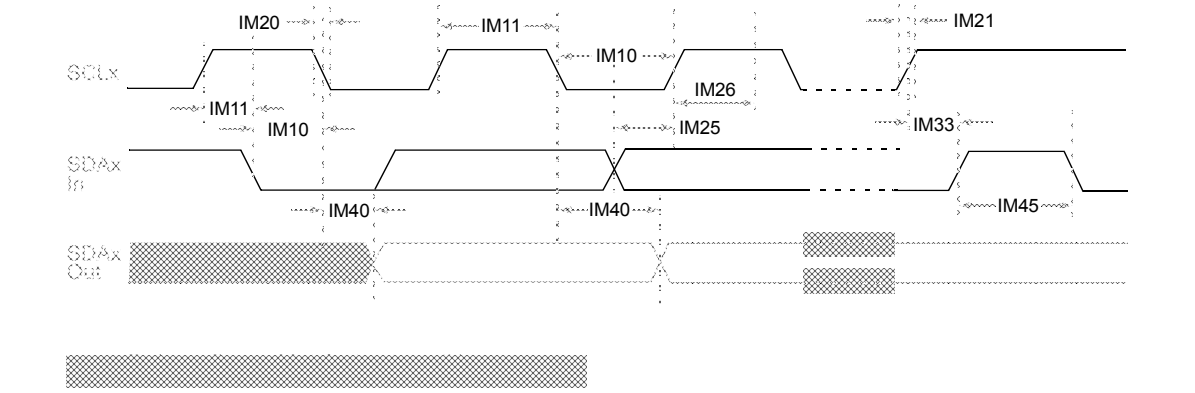


TABLE 30-46: I2Cx BUS DATA TIMING REQUIREMENTS (MASTER MODE)

AC CHARACTERISTICS				Standard Operating Conditions: 4.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Param No.	**Symbol**	**Characteristic(4)**		**Min.(1)**	**Max.**	**Units**	**Conditions**
IM10	TLO:SCL	Clock Low Time	100 kHz mode	TCY/2 (BRG + 2)	—	µs	
			400 kHz mode	TCY/2 (BRG + 2)	—	µs	
			1 MHz mode(2)	TCY/2 (BRG + 2)	—	µs	
IM11	THI:SCL	Clock High Time	100 kHz mode	TCY/2 (BRG + 2)	—	µs	
			400 kHz mode	TCY/2 (BRG + 2)	—	µs	
			1 MHz mode(2)	TCY/2 (BRG + 2)	—	µs	
IM20	TF:SCL	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode(2)	—	100	ns	
IM21	TR:SCL	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode(2)	—	300	ns	
IM25	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	
			400 kHz mode	100	—	ns	
			1 MHz mode(2)	40	—	ns	
IM26	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	µs	
			400 kHz mode	0	0.9	µs	
			1 MHz mode(2)	0.2	—	µs	
IM30	TSU:STA	Start Condition Setup Time	100 kHz mode	TCY/2 (BRG + 2)	—	µs	Only relevant for Repeated Start condition
			400 kHz mode	TCY/2 (BRG + 2)	—	µs	
			1 MHz mode(2)	TCY/2 (BRG + 2)	—	µs	
IM31	THD:STA	Start Condition Hold Time	100 kHz mode	TCY/2 (BRG + 2)	—	µs	After this period, the first clock pulse is generated
			400 kHz mode	TCY/2 (BRG +2)	—	µs	
			1 MHz mode(2)	TCY/2 (BRG + 2)	—	µs	
IM33	TSU:STO	Stop Condition Setup Time	100 kHz mode	TCY/2 (BRG + 2)	—	µs	
			400 kHz mode	TCY/2 (BRG + 2)	—	µs	
			1 MHz mode(2)	TCY/2 (BRG + 2)	—	µs	
IM34	THD:STO	Stop Condition Hold Time	100 kHz mode	TCY/2 (BRG + 2)	—	µs	
			400 kHz mode	TCY/2 (BRG + 2)	—	µs	
			1 MHz mode(2)	TCY/2 (BRG + 2)	—	µs	
IM40	TAA:SCL	Output Valid From Clock	100 kHz mode	—	3500	ns	
			400 kHz mode	—	1000	ns	
			1 MHz mode(2)	—	400	ns	
IM45	TBF:SDA	Bus Free Time	100 kHz mode	4.7	—	µs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	µs	
			1 MHz mode(2)	0.5	—	µs	
IM50	CB	Bus Capacitive Loading		—	400	pF	
IM51	TPGD	Pulse Gobbler Delay		65	390	ns	See **Note 3**

Note 1: BRG is the value of the I^2C Baud Rate Generator. Refer to **"Inter-Integrated Circuit™ (I^2C™)"** (DS70000195) in the *"dsPIC33/PIC24 Family Reference Manual"*. Please see the Microchip web site for the latest *"dsPIC33/PIC24 Family Reference Manual"* sections.

2: Maximum pin capacitance = 10 pF for all I2Cx pins (for 1 MHz mode only).

3: Typical value for this parameter is 130 ns.

4: These parameters are characterized but not tested in manufacturing.

FIGURE 32-47: TYPICAL INL (V_{DD} = 5.5V, +85°C)

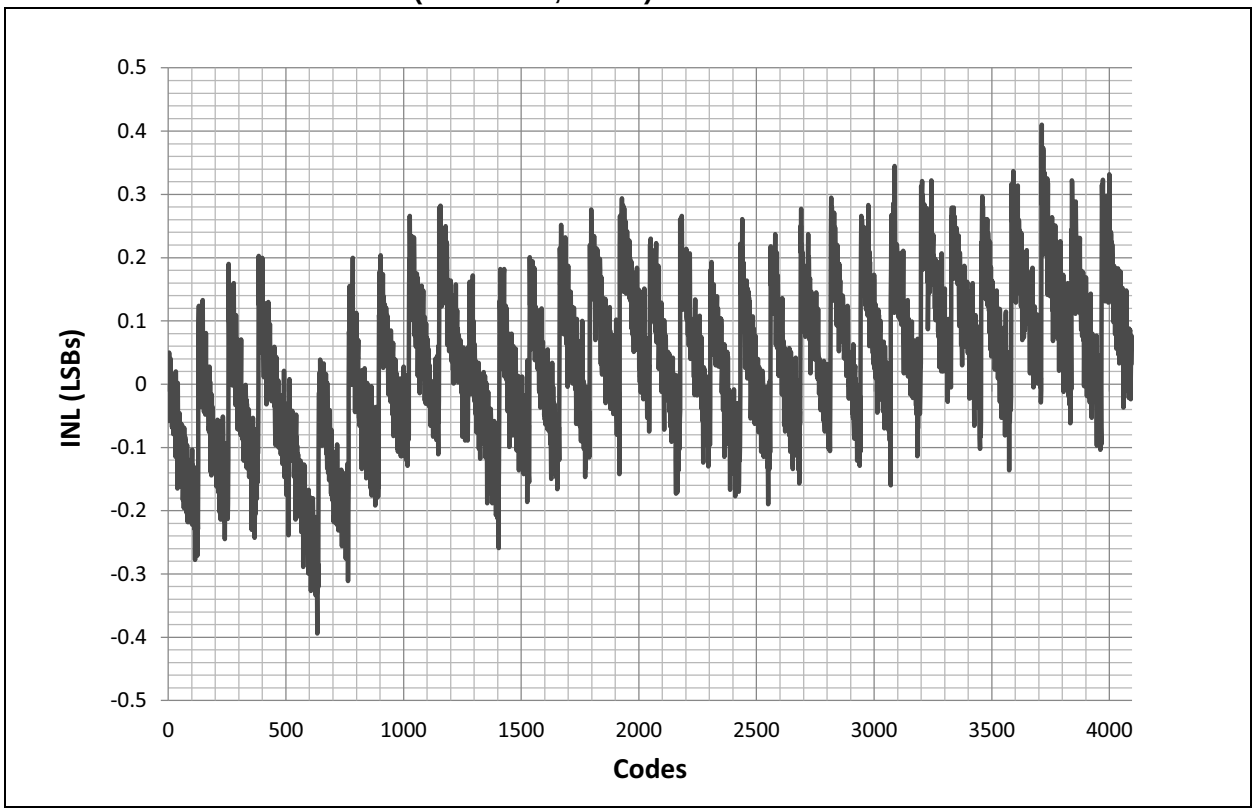
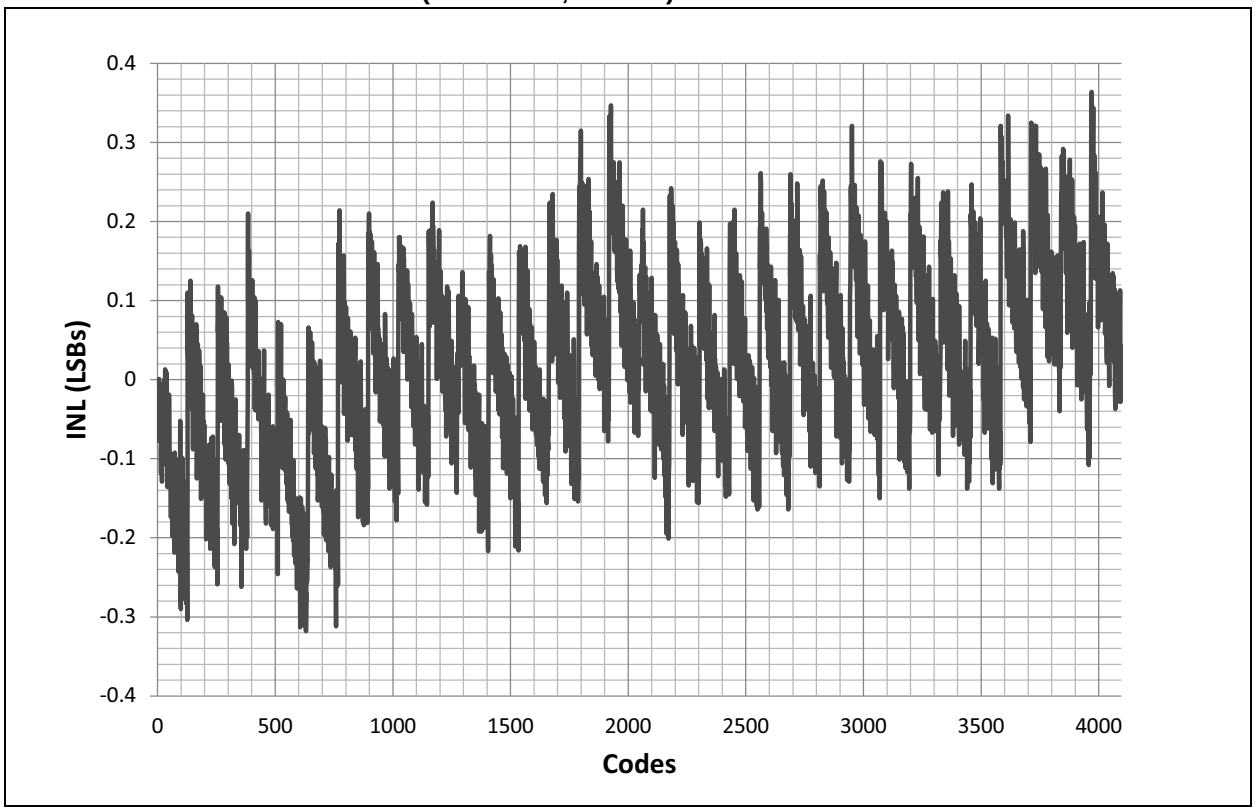


FIGURE 32-48: TYPICAL INL (V_{DD} = 5.5V, +125°C)



33.2 IIDLE

FIGURE 33-5: TYPICAL/MAXIMUM IIDLE vs. FOSC (EC MODE 10 MHz TO 40 MHz, 5.5V MAX)

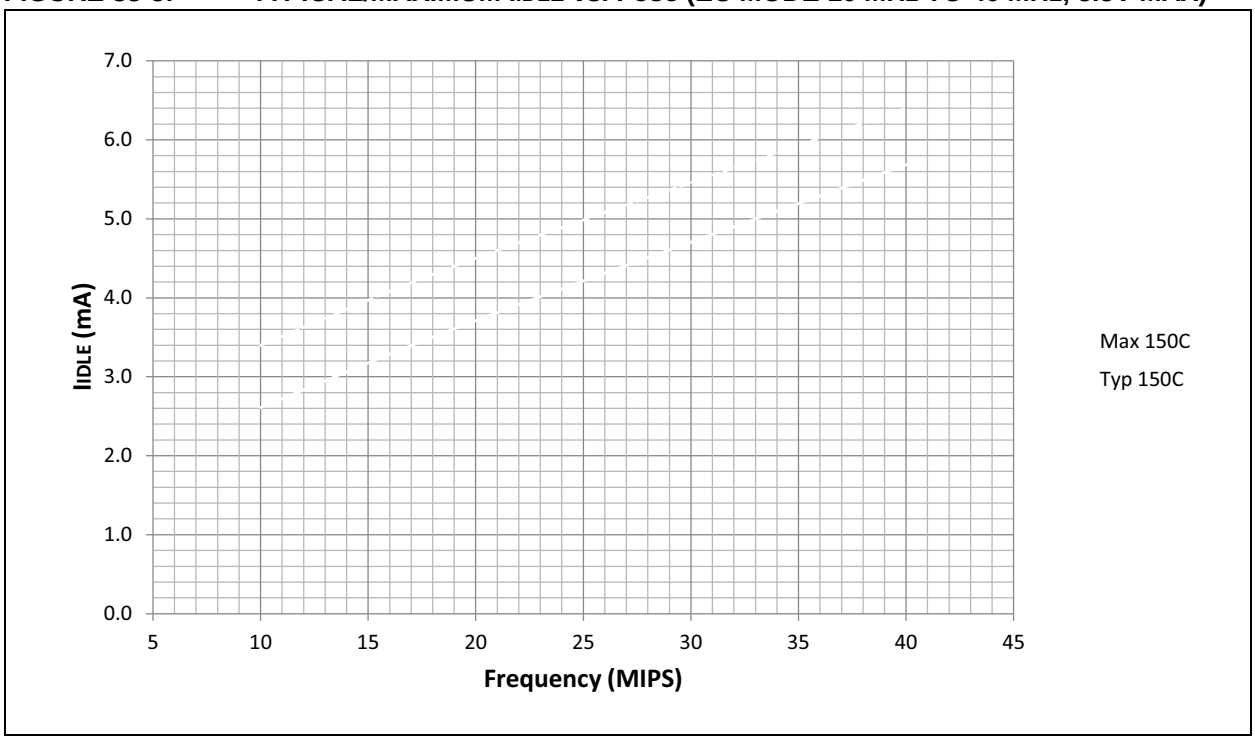


FIGURE 33-6: TYPICAL IIDLE vs. VDD (EC MODE, 10 MIPS)