



Computer Science

Fall 2024: CSCI 181RT Real-Time Systems in the Real World

Lecture 21

Tuesday, November 7, 2024
Edmunds Hall 105
2:45 PM - 4:00 PM

Professor Jennifer DesCombes

Agenda

- Go Backs
- Discussion on Assignments
- Lab #10 Review
- Fielder Choice [FC 6-4] - Various Hardware Topics
- Look Ahead
- Assignment
- Action Items

Go Backs

- General?
- Action Item Status
 - AI240910-2: Find recommended book on computer architecture.
 - AI240924-1: At what point as a development team grows does it make sense to have dedicated software and integration testers?
 - AI241024-1: Provide documentation on how to disable compiler optimization.

Discussion on Assignments - Part 1

- The Soul Of A New Machine
 - **Not Due Until Next Tuesday**
- Chip Documentation - Chapter 17 - Input Capture
 - Functional Modes
 - Register Operations
 - Download Full Documentation

Hardware Features and Capabilities - OS Related

- Input Capture - Up to 9 Circuits per Package

17.0 INPUT CAPTURE

Note: This data sheet summarizes the features of the PIC32MZ EF family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 15. “Input Capture”** (DS60001122) of the “*PIC32 Family Reference Manual*”, which is available from the Microchip web site (www.microchip.com/PIC32).

The Input Capture module is useful in applications requiring frequency (period) and pulse measurement.

The Input Capture module captures the 16-bit or 32-bit value of the selected Time Base registers when an event occurs at the ICx pin.

Capture events are caused by the following:

- Capture timer value on every edge (rising and falling), specified edge first

- Prescaler capture event modes:
 - Capture timer value on every 4th rising edge of input at ICx pin
 - Capture timer value on every 16th rising edge of input at ICx pin

Each input capture channel can select between one of six 16-bit timers for the time base, or two of six 16-bit timers together to form a 32-bit timer. The selected timer can use either an internal or external clock.

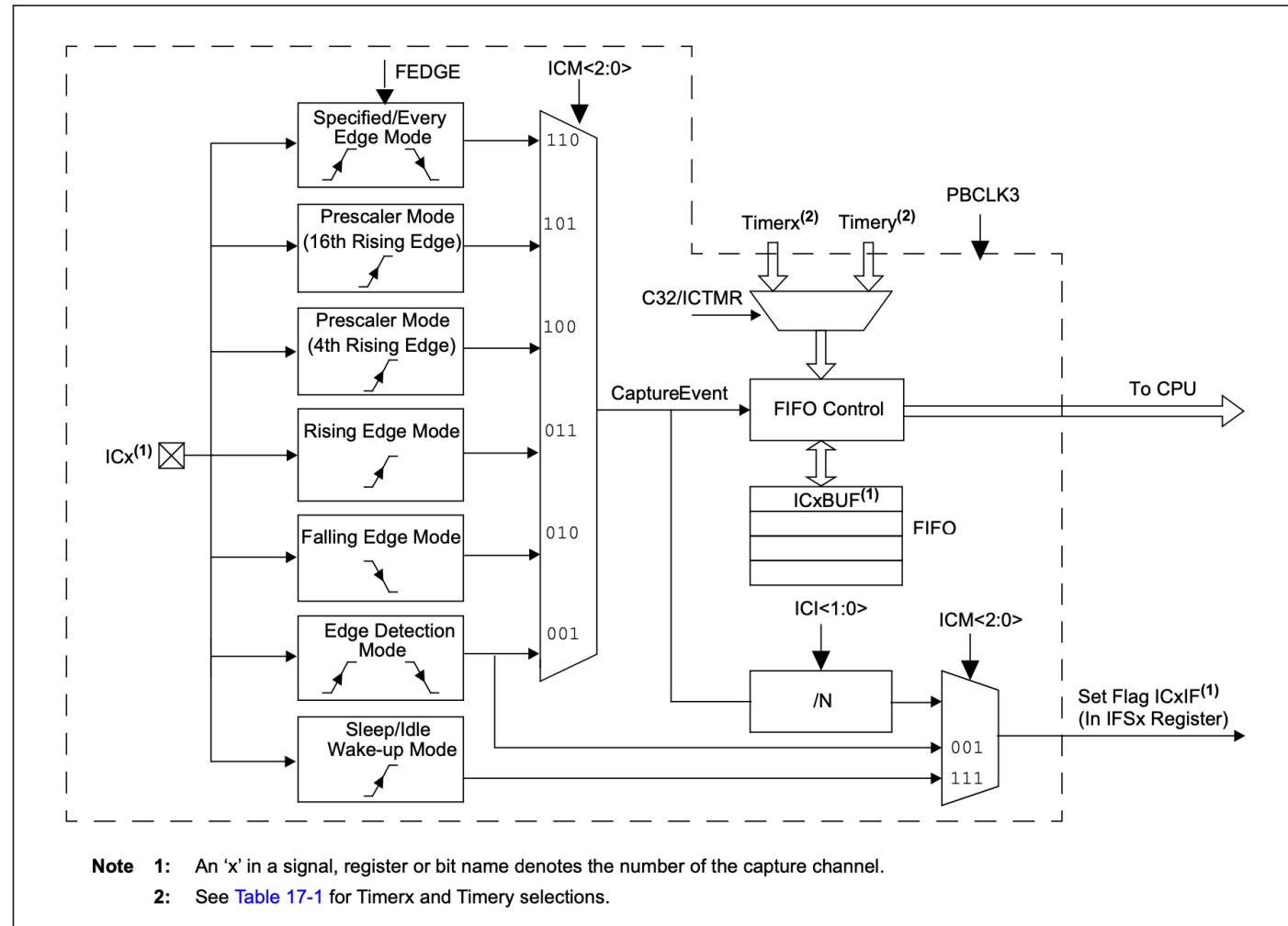
Other operational features include:

- Device wake-up from capture pin during Sleep and Idle modes
- Interrupt on input capture event
- 4-word FIFO buffer for capture values; Interrupt optionally generated after 1, 2, 3, or 4 buffer locations are filled
- Input capture can also be used to provide additional sources of external interrupts

Hardware Features and Capabilities - OS Related

- Input Capture

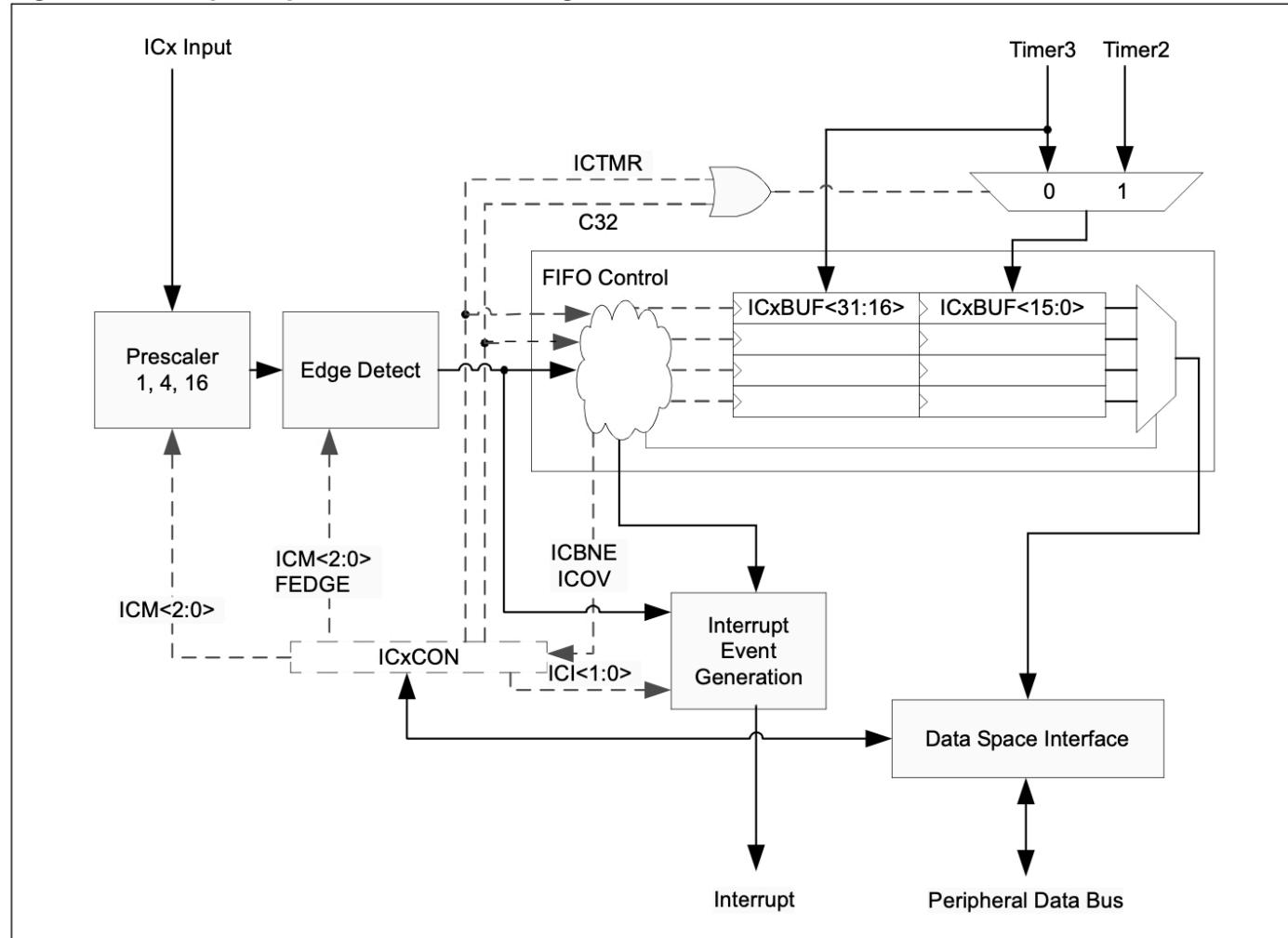
FIGURE 17-1: INPUT CAPTURE BLOCK DIAGRAM



Hardware Features and Capabilities - OS Related

- Input Capture from PIC32 Family Reference Manual

Figure 15-1: Input Capture Module Block Diagram



Hardware Features and Capabilities - OS Related

- Input Capture from PIC32 Family Reference Manual

Register 15-1: ICxCON: Input Capture 'x' Control Register

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
	ON	—	SIDL	—	—	—	FEDGE	C32
7:0	R/W-0	R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0
	ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>		

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

Hardware Features and Capabilities - OS Related

- Input Capture from PIC32 Family Reference Manual

bit 31-16 **Unimplemented:** Read as '0'

bit 15 **ON:** Input Capture Module Enable bit

1 = Module is enabled

0 = Disable and reset module, disable clocks, disable interrupt generation and allow SFR modifications

Note: When using 1:1 PBCLK divisor, the user's software should not read/write the peripheral's SFRs in the SYCLK cycle immediately following the instruction that clears the module's ON bit.

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

bit 13 **SIDL:** Stop in Idle Control bit

1 = Halt in CPU Idle mode

0 = Continue to operate in CPU Idle mode

bit 12-10 **Unimplemented:** Read as '0'

bit 9 **FEDGE:** First Capture Edge Select bit (only used in mode 6, ICM<2:0> = 110)

1 = Capture rising edge first

0 = Capture falling edge first

bit 8 **C32:** 32-bit Capture Select bit

1 = 32-bit timer resource capture

0 = 16-bit timer resource capture

bit 7 **ICTMR:** Timer Select bit (Does not affect timer selection when C32 (ICxCON<8>) is '1')

0 = Timer3 is the counter source for capture

1 = Timer2 is the counter source for capture

bit 6-5 **ICI<1:0>:** Interrupt Control bits

11 = Interrupt on every fourth capture event

10 = Interrupt on every third capture event

01 = Interrupt on every second capture event

00 = Interrupt on every capture event

bit 4 **ICOV:** Input Capture Overflow Status Flag bit (read-only)

1 = Input capture overflow occurred

0 = No input capture overflow occurred

bit 3 **ICBNE:** Input Capture Buffer Not Empty Status bit (read-only)

1 = Input capture buffer is not empty; at least one more capture value can be read

0 = Input capture buffer is empty

bit 2-0 **ICM<2:0>:** Input Capture Mode Select bits

111 = Interrupt-Only mode (only supported while in Sleep mode or Idle mode)

110 = Simple Capture Event mode – every edge, specified edge first and every edge thereafter

101 = Prescaled Capture Event mode – every sixteenth rising edge

100 = Prescaled Capture Event mode – every fourth rising edge

011 = Simple Capture Event mode – every rising edge

010 = Simple Capture Event mode – every falling edge

001 = Edge Detect mode – every edge (rising and falling)

000 = Capture Disable mode

Discussion on Assignments - Part 2

- Chip Documentation - Chapter 21 - UART
 - Review Block Diagrams
 - Look for Interrupt Options

Hardware Features and Capabilities - OS Related

- Serials Port - UART - Up to 6 Circuits per Package

22.0 UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

Note: This data sheet summarizes the features of the PIC32MZ EF family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 21. “Universal Asynchronous Receiver Transmitter (UART)”** (DS60001107) in the “*PIC32 Family Reference Manual*”, which is available from the Microchip web site (www.microchip.com/PIC32).

The UART module is one of the serial I/O modules available in the PIC32MZ EF family of devices. The UART is a full-duplex, asynchronous communication channel that communicates with peripheral devices and personal computers through protocols, such as RS-232, RS-485, LIN, and IrDA®. The module also supports the hardware flow control option, with UxCTS and UxRTS pins, and also includes an IrDA encoder and decoder.

The primary features of the UART module are:

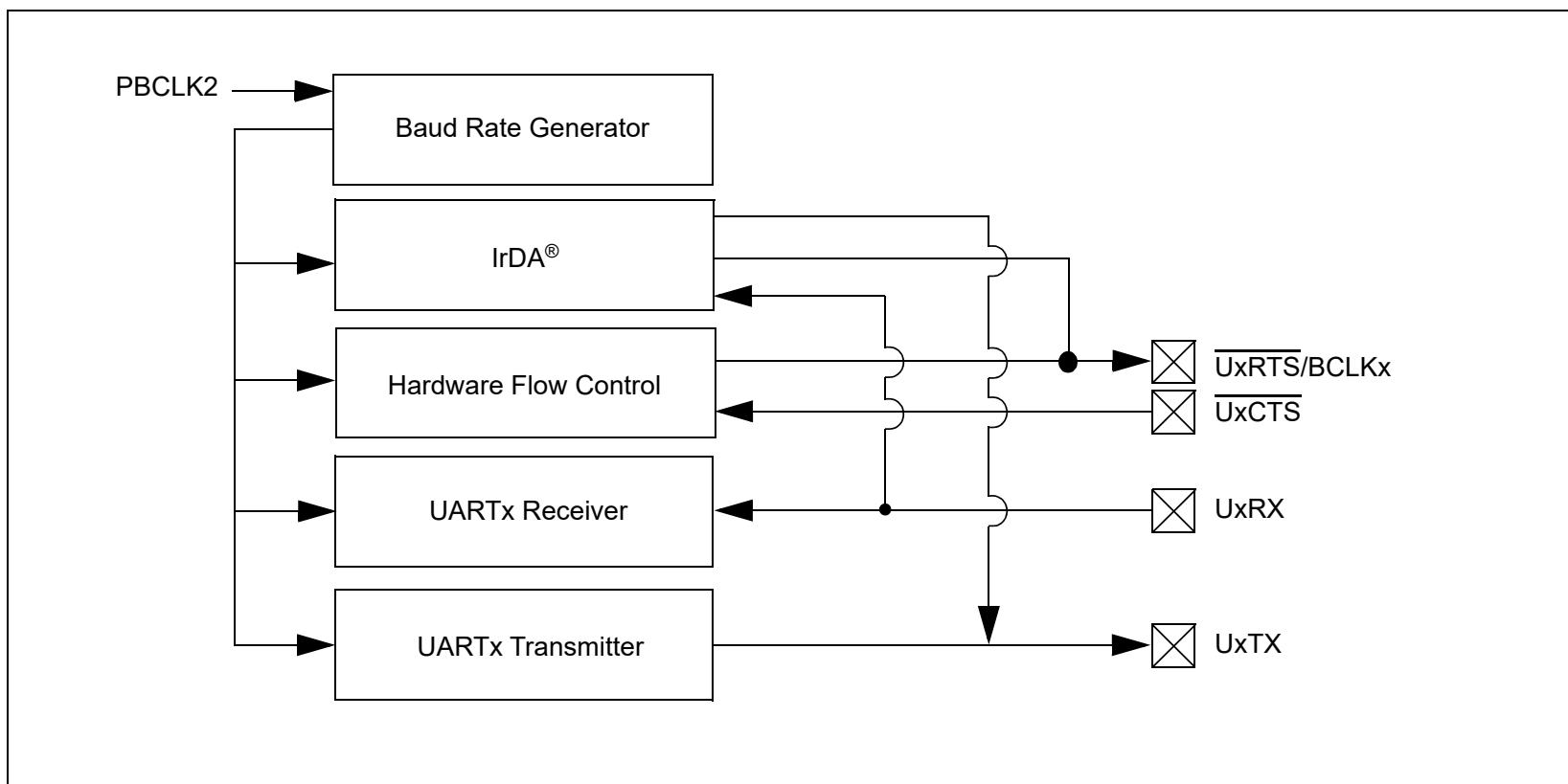
- Full-duplex, 8-bit or 9-bit data transmission
- Even, Odd or No Parity options (for 8-bit data)
- One or two Stop bits
- Hardware auto-baud feature
- Hardware flow control option
- Fully integrated Baud Rate Generator (BRG) with 16-bit prescaler
- Baud rates ranging from 76 bps to 25 Mbps at 100 MHz (PBCLK2)
- 8-level deep First-In-First-Out (FIFO) transmit data buffer
- 8-level deep FIFO receive data buffer
- Parity, framing and buffer overrun error detection
- Support for interrupt-only on address detect (9th bit = 1)
- Separate transmit and receive interrupts
- Loopback mode for diagnostic support
- LIN Protocol support
- IrDA encoder and decoder with 16x baud clock output for external IrDA encoder/decoder support

Figure 22-1 illustrates a simplified block diagram of the UART module.

Hardware Features and Capabilities - OS Related

- UART

FIGURE 22-1: UART SIMPLIFIED BLOCK DIAGRAM

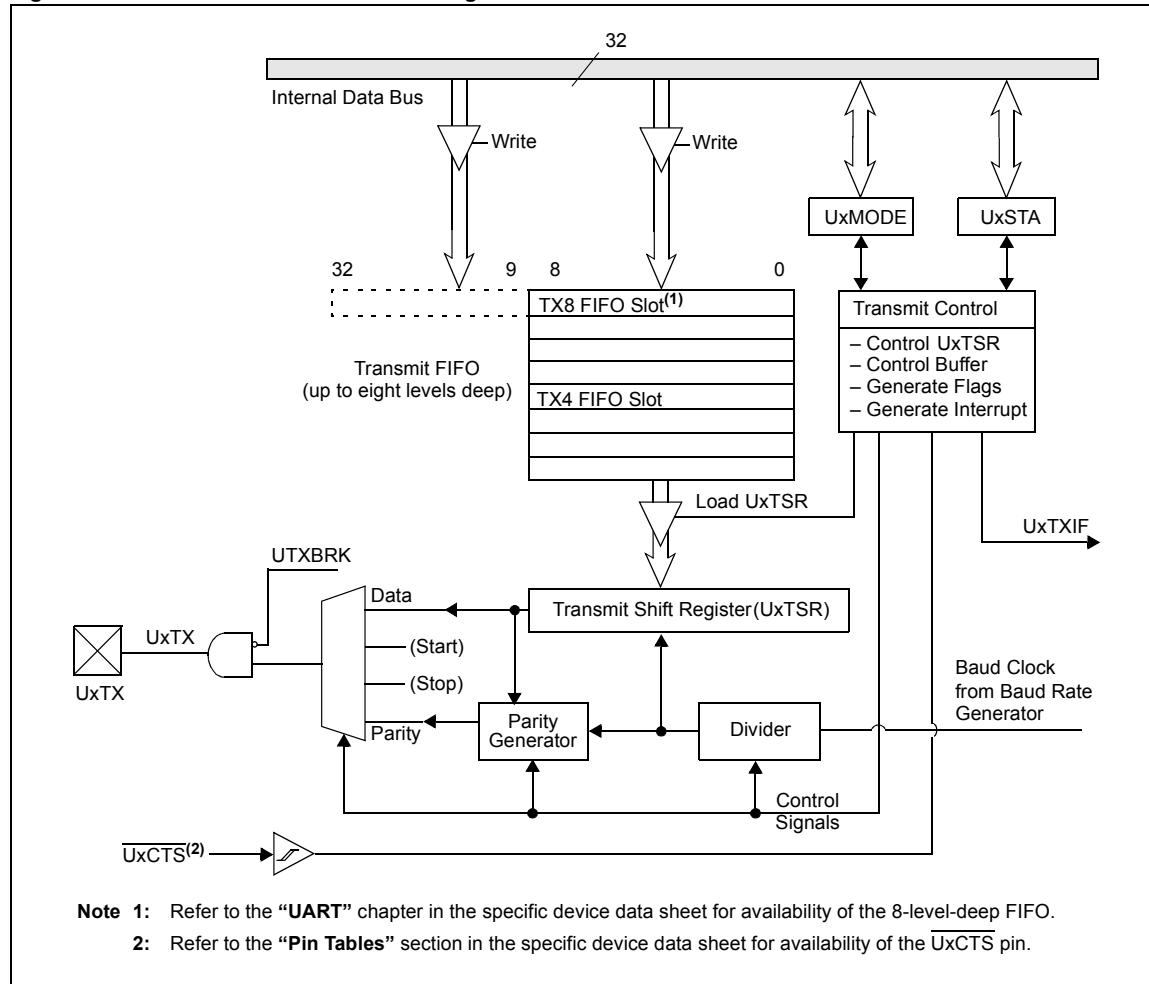




Hardware Features and Capabilities - OS Related

- UART - Detailed Tx Block Diagram from PIC32 Family Reference Manual

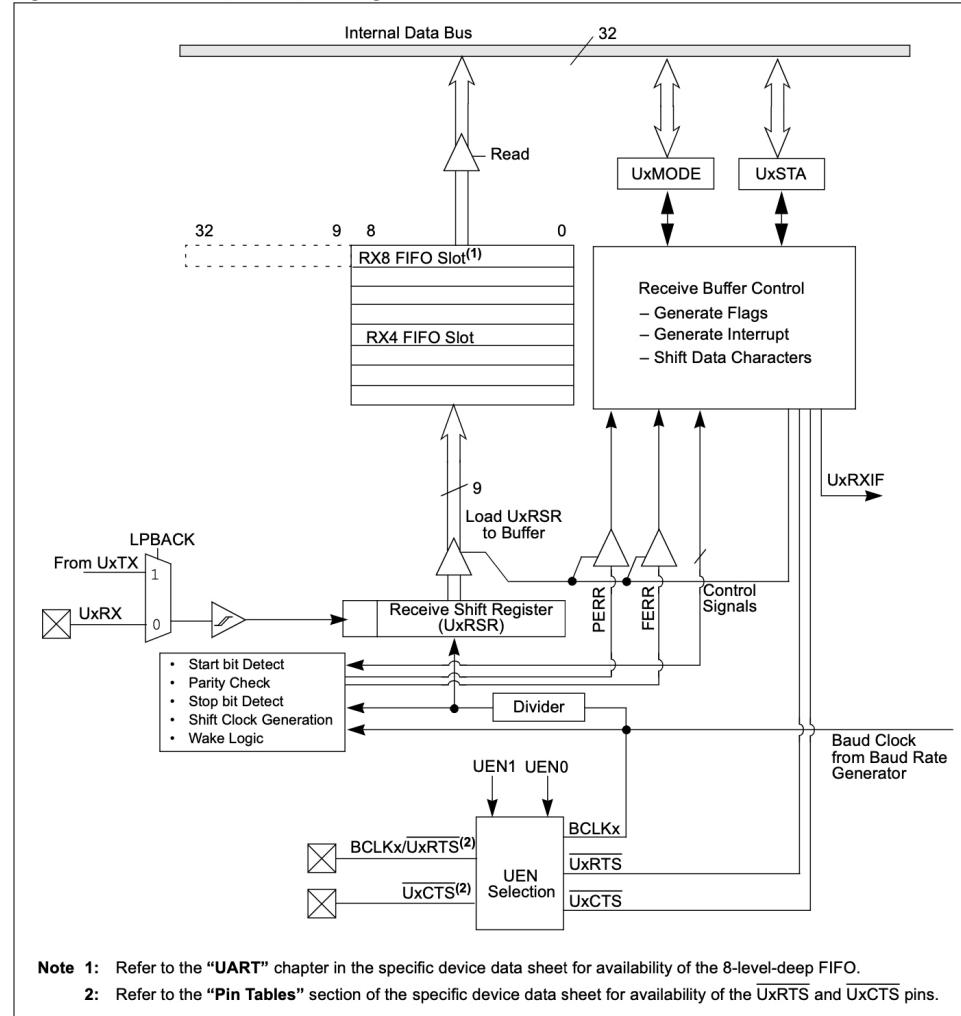
Figure 21-3: UART Transmitter Block Diagram



Hardware Features and Capabilities - OS Related

- UART - Detailed Rx Block Diagram from PIC32 Family Reference Manual

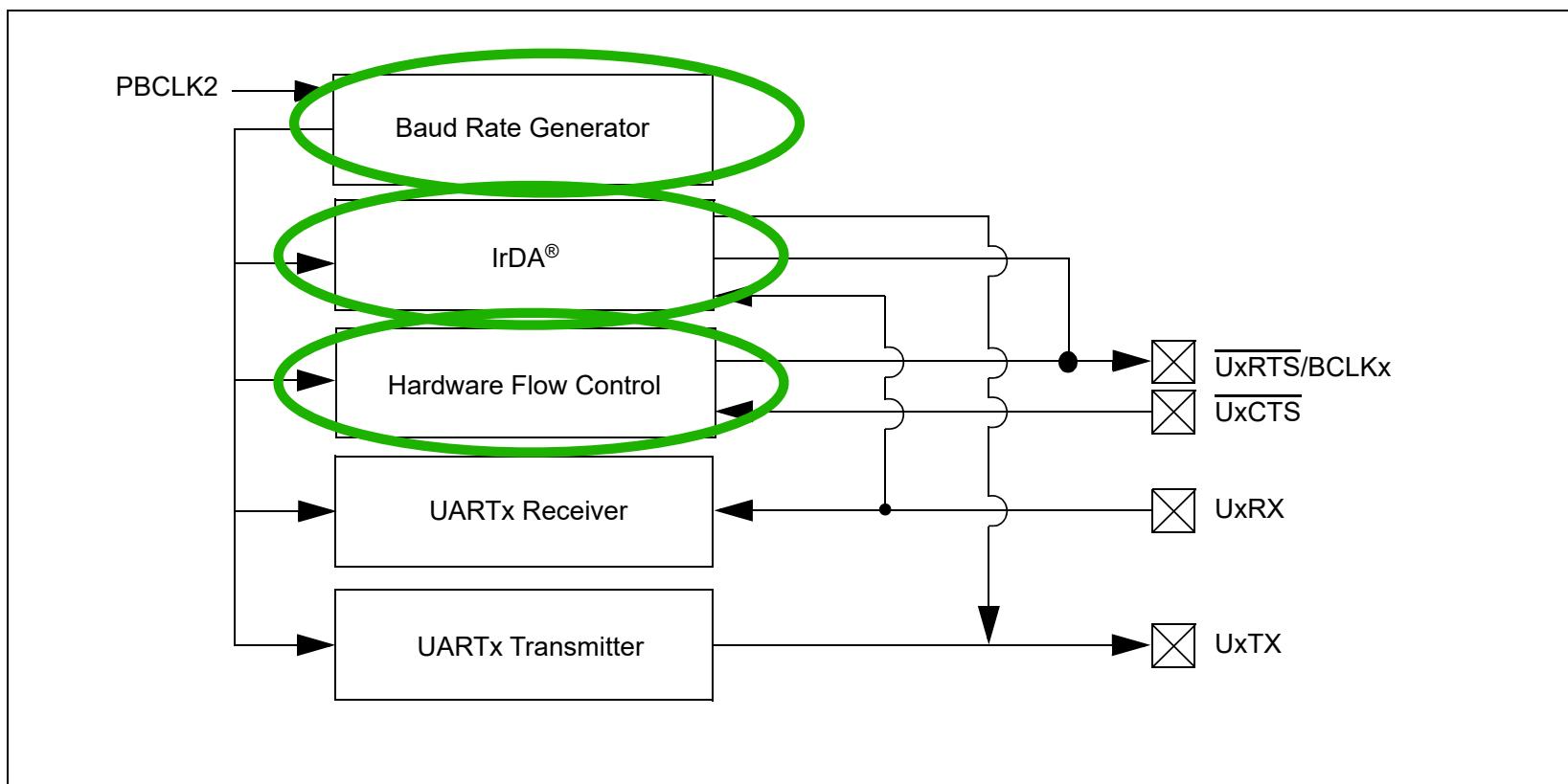
Figure 21-7: UART Receiver Block Diagram



Hardware Features and Capabilities - OS Related

- UART

FIGURE 22-1: UART SIMPLIFIED BLOCK DIAGRAM



Lab #10 Review

- Goals for Prior Labs
 - Read Digital Input (GPIO1, Connector 501-Pin 5, Processor RK4) ✓
 - Drive LED to Match Digital Input ✓
 - Use Function Generator to Experiment ✓
- Integration of New FreeRTOS Framework ✓
 - Good Questions Asked
 - Suggestions Being Incorporated into Framework

Fielder Choice [FC 6-4] - Various Hardware Topics

- Runner on 1st, Batter Hits Ball to Short Stop (#6), Short Stop Throws to Second Baseman (#4) for a Force Out at Second.
- Various Hardware Topics
 - Memory Organizations and Permissions
 - Flash Program Memory with Live Program Updates
 - Interrupts and Interrupt Vector Tables

Memory Organization

- User Address Space
- Kernel Address Space
- Boot Flash
- Read/Write Permissions
- Extern Memory Mapped into 4GB Virtual Space

4.0 MEMORY ORGANIZATION

Note: This data sheet summarizes the features of the PIC32MZ EF family of devices. It is not intended to be a comprehensive reference source. For detailed information, refer to **Section 48. "Memory Organization and Permissions"** in the "*PIC32 Family Reference Manual*", which is available from the Microchip web site (www.microchip.com/PIC32).

PIC32MZ EF microcontrollers provide 4 GB of unified virtual memory address space. All memory regions, including program, data memory, SFRs and Configuration registers, reside in this address space at their respective unique addresses. The program and data memories can be optionally partitioned into user and kernel memories. In addition, PIC32MZ EF devices allow execution from data memory.

Key features include:

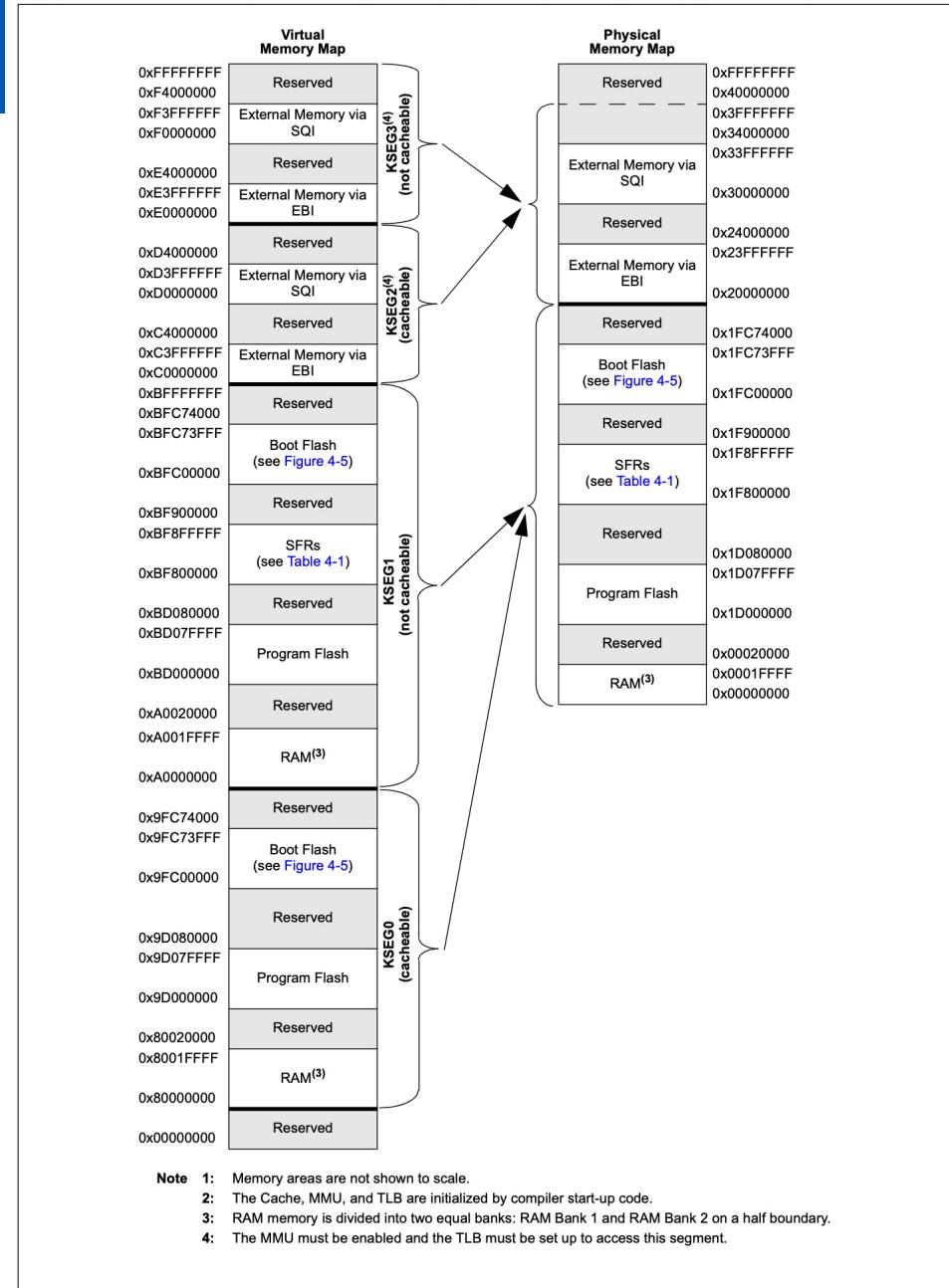
- 32-bit native data width
- Separate User (KUSEG) and Kernel (KSEG0/KSEG1/KSEG2/KSEG3) mode address space
- Separate boot Flash memory for protected code
- Robust bus exception handling to intercept runaway code
- Cacheable (KSEG0/KSEG2) and non-cacheable (KSEG1/KSEG3) address regions
- Read/write permission access to predefined memory regions



Memory Organization

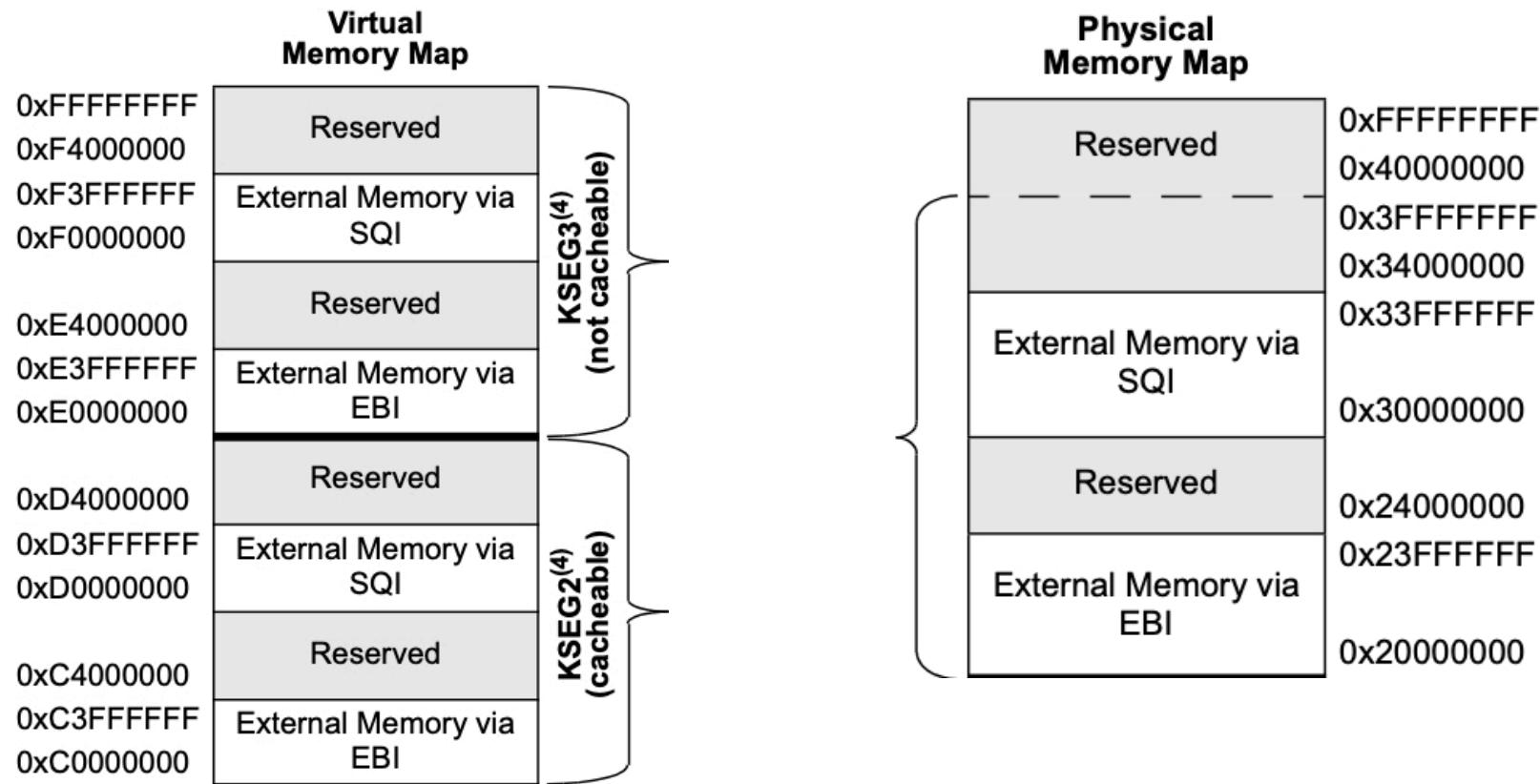
- Virtual Memory - 4GB
- Physical Memory
- Hardware Performs Mapping from Virtual to Physical Based on Complier and Linker Directives

FIGURE 4-1: MEMORY MAP FOR DEVICES WITH 512 KB OF PROGRAM MEMORY^(1,2)



Memory Organization

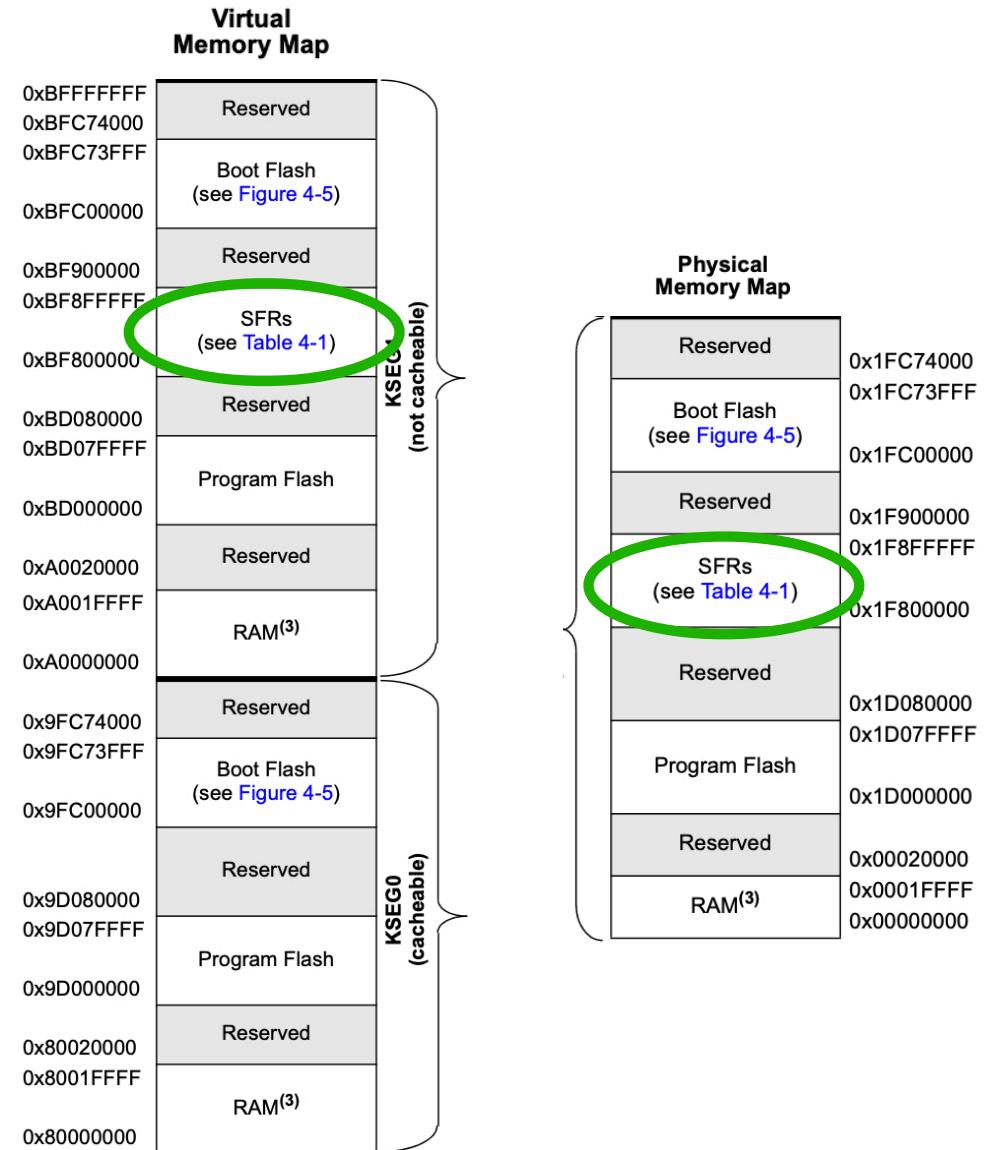
- External Memory Options





Memory Organization

- Internal Memory Options
- Includes Special Function Registers (SFRs)
- Protected Space is Controlled by Virtual Address
- Map Varies by RAM and Flash Capacity



Flash Program Memory

- Section 52. Flash Memory with Support for Live Update

52.1 INTRODUCTION

This document describes techniques for programming the Flash memory on PIC32 devices that support Live Update. These devices contain up to two banks of Flash memory each with their own Boot Flash Memory (BFM) partition and Program Flash Memory (PFM) partition for storing user code or non-volatile data. The dual memory bank feature allows Flash to be programmed in one bank while executing from another for live updates of program memory. There are three methods by which the user can program this memory:

- Run-Time Self-Programming (RTSP) – performed by the user's software
- In-Circuit Serial Programming™ (ICSP™) – performed using a serial data connection to the device, which allows much faster programming than RTSP
- Enhanced Joint Test Action Group Programming (EJTAG) – performed by an EJTAG-capable programmer, using the EJTAG port of the device

Physical Memory Map	
Reserved	0x1FC74000
Boot Flash (see Figure 4-5)	0x1FC73FFF
Reserved	0x1FC00000
SFRs (see Table 4-1)	0x1F900000
Reserved	0x1F8FFFFF
Reserved	0x1F800000
Program Flash	0x1D080000
Reserved	0x1D07FFFF
RAM ⁽³⁾	0x1D000000
Reserved	0x00020000
RAM ⁽³⁾	0x0001FFFF
RAM ⁽³⁾	0x00000000

Flash Program Memory

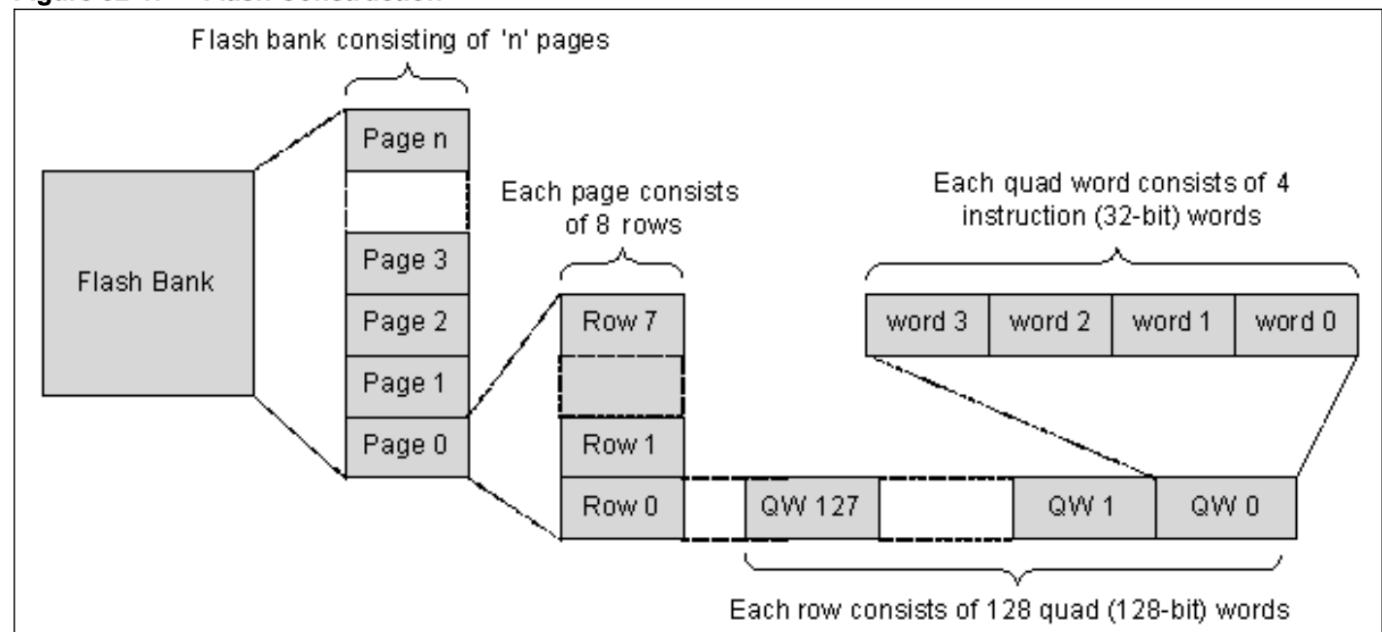
- Section 52. Flash Memory Organization

Each bank of Flash memory is divided into pages. A page is the smallest unit of memory that can be erased at one time.

Each page of memory is segmented into eight rows. A row is the largest unit of memory that can be programmed at one time. A row consists of 128 Quad (128-bit) words.

Each Quad word consists of four instruction (32-bit) words. Flash memory can be programmed in rows, Quad word (128-bit) or Word (32-bit) units.

Figure 52-1: Flash Construction



Interrupts and Interrupt Vector Tables

- Up to 213 Interrupt Sources

7.0 CPU EXCEPTIONS AND INTERRUPT CONTROLLER

Note: This data sheet summarizes the features of the PIC32MZ EF family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 8. “Interrupt Controller”** (DS60001108) and **Section 50. “CPU for Devices with MIPS32® microAptiv™ and M-Class Cores”** (DS60001192) of the “*PIC32 Family Reference Manual*”, which is available from the Microchip web site (www.microchip.com/PIC32).

PIC32MZ EF devices generate interrupt requests in response to interrupt events from peripheral modules. The Interrupt Controller module exists outside of the CPU and prioritizes the interrupt events before presenting them to the CPU.

The CPU handles interrupt events as part of the exception handling mechanism, which is described in [Section 7.1 “CPU Exceptions”](#).

The Interrupt Controller module includes the following features:

- Up to 213 interrupt sources and vectors with dedicated programmable offsets, eliminating the need for redirection
- Single and multi-vector mode operations
- Five external interrupts with edge polarity control
- Interrupt proximity timer
- Seven user-selectable priority levels for each vector
- Four user-selectable subpriority levels within each priority
- Seven shadow register sets that can be used for any priority level, eliminating software context switch and reducing interrupt latency
- Software can generate any interrupt

[Figure 7-1](#) shows the block diagram for the Interrupt Controller and CPU exceptions.

Interrupts and Interrupt Vector Tables

- Each Interrupt Fully Configurable
- Offset Spacing of Interrupts Allows Custom Features

7.2 Interrupts

The PIC32MZ EF family uses variable offsets for vector spacing. This allows the interrupt vector spacing to be configured according to application needs. A unique interrupt vector offset can be set for each vector using its associated OFFx register.

For details on the Variable Offset feature, refer to **8.5.2 “Variable Offset”** in **Section 8. “Interrupt Controller”** (DS60001108) of the **“PIC32 Family Reference Manual”**.

[Table 7-2](#) provides the Interrupt IRQ, vector and bit location information.

TABLE 7-2: INTERRUPT IRQ, VECTOR, AND BIT LOCATION

Interrupt Source ⁽¹⁾	XC32 Vector Name	IRQ #	Vector #	Interrupt Bit Location				Persistent Interrupt
				Flag	Enable	Priority	Sub-priority	
Highest Natural Order Priority								
Core Timer Interrupt	_CORE_TIMER_VECTOR	0	OFF000<17:1>	IFS0<0>	IEC0<0>	IPC0<4:2>	IPC0<1:0>	No
Core Software Interrupt 0	_CORE_SOFTWARE_0_VECTOR	1	OFF001<17:1>	IFS0<1>	IEC0<1>	IPC0<12:10>	IPC0<9:8>	No
Core Software Interrupt 1	_CORE_SOFTWARE_1_VECTOR	2	OFF002<17:1>	IFS0<2>	IEC0<2>	IPC0<20:18>	IPC0<17:16>	No
External Interrupt 0	_EXTERNAL_0_VECTOR	3	OFF003<17:1>	IFS0<3>	IEC0<3>	IPC0<28:26>	IPC0<25:24>	No
Timer1	_TIMER_1_VECTOR	4	OFF004<17:1>	IFS0<4>	IEC0<4>	IPC1<4:2>	IPC1<1:0>	No
Input Capture 1 Error	_INPUT_CAPTURE_1_ERROR_VECTOR	5	OFF005<17:1>	IFS0<5>	IEC0<5>	IPC1<12:10>	IPC1<9:8>	Yes
Input Capture 1	_INPUT_CAPTURE_1_VECTOR	6	OFF006<17:1>	IFS0<6>	IEC0<6>	IPC1<20:18>	IPC1<17:16>	Yes
Output Compare 1	_OUTPUT_COMPARE_1_VECTOR	7	OFF007<17:1>	IFS0<7>	IEC0<7>	IPC1<28:26>	IPC1<25:24>	No
External Interrupt 1	_EXTERNAL_1_VECTOR	8	OFF008<17:1>	IFS0<8>	IEC0<8>	IPC2<4:2>	IPC2<1:0>	No
Timer2	_TIMER_2_VECTOR	9	OFF009<17:1>	IFS0<9>	IEC0<9>	IPC2<12:10>	IPC2<9:8>	No

Interrupts and Interrupt Vector Tables

- Yes, the Table Actually Has 213 Entries!

ADC1 Early Interrupt	_ADC1_EARLY_VECTOR	199	OFF199<17:1>	IFS6</>	IEC6</>	IPC49<28:26>	IPC49<25:24>	Yes
ADC2 Early Interrupt	_ADC2_EARLY_VECTOR	200	OFF200<17:1>	IFS6<8>	IEC6<8>	IPC50<4:2>	IPC50<1:0>	Yes
ADC3 Early Interrupt	_ADC2_EARLY_VECTOR	201	OFF201<17:1>	IFS6<9>	IEC6<9>	IPC50<12:10>	IPC50<9:8>	Yes
ADC4 Early Interrupt	_ADC4_EARLY_VECTOR	202	OFF202<17:1>	IFS6<10>	IEC6<10>	IPC50<20:18>	IPC50<17:16>	Yes
Reserved	—	203	—	—	—	—	—	—
Reserved	—	204	—	—	—	—	—	—
ADC7 Early Interrupt	_ADC7_EARLY_VECTOR	205	OFF205<17:1>	IFS6<13>	IEC6<13>	IPC51<12:10>	IPC51<9:8>	Yes
ADC0 Warm Interrupt	_ADC0_WARM_VECTOR	206	OFF206<17:1>	IFS6<14>	IEC6<14>	IPC51<20:18>	IPC51<17:16>	Yes
ADC1 Warm Interrupt	_ADC1_WARM_VECTOR	207	OFF207<17:1>	IFS6<15>	IEC6<15>	IPC51<28:26>	IPC51<25:24>	Yes
ADC2 Warm Interrupt	_ADC2_WARM_VECTOR	208	OFF208<17:1>	IFS6<16>	IEC6<16>	IPC52<4:2>	IPC52<1:0>	Yes
ADC3 Warm Interrupt	_ADC3_WARM_VECTOR	209	OFF209<17:1>	IFS6<17>	IEC6<17>	IPC52<12:10>	IPC52<9:8>	Yes
ADC4 Warm Interrupt	_ADC4_WARM_VECTOR	210	OFF210<17:1>	IFS6<18>	IEC6<18>	IPC52<20:18>	IPC52<17:16>	Yes
Reserved	—	211	—	—	—	—	—	—
Reserved	—	212	—	—	—	—	—	—
ADC7 Warm Interrupt	_ADC7_WARM_VECTOR	213	OFF213<17:1>	IFS6<21>	IEC6<21>	IPC53<12:10>	IPC53<9:8>	Yes

Lowest Natural Order Priority

Note 1: Not all interrupt sources are available on all devices. See [TABLE 1: "PIC32MZ EF Family Features"](#) for the list of available peripherals.

2: This interrupt source is not available on 64-pin devices.

3: This interrupt source is not available on 100-pin devices.

4: This interrupt source is not available on 124-pin devices.

Interrupts and Interrupt Vector Tables

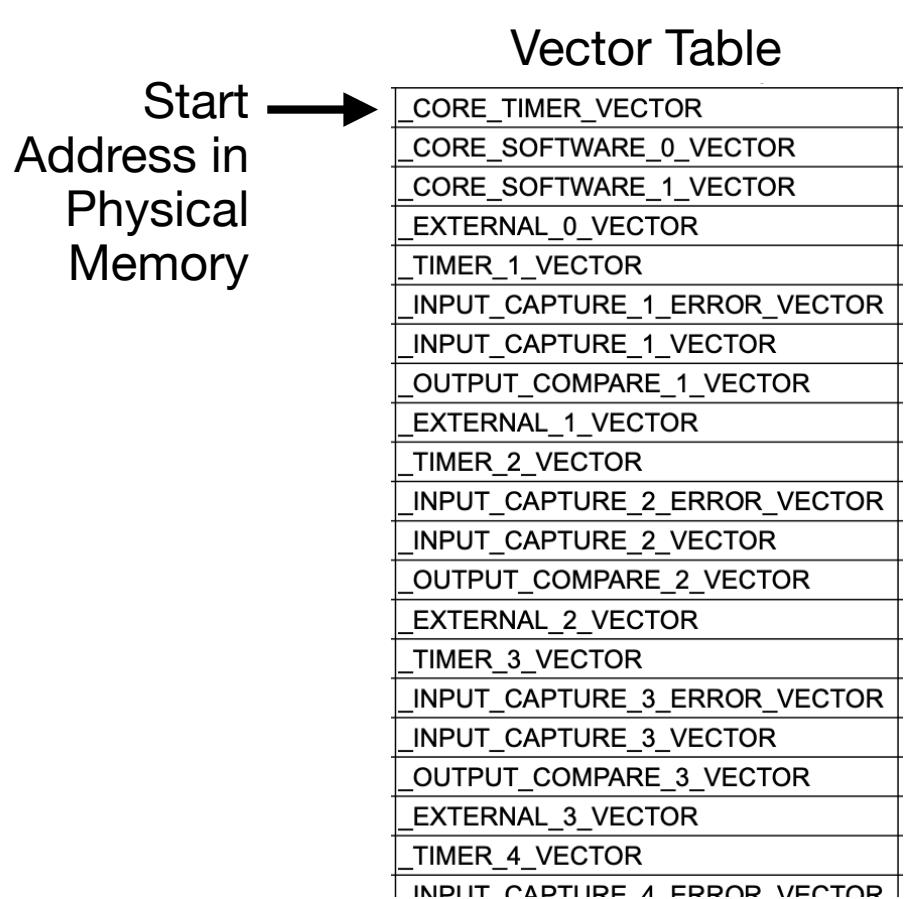
- First Portion of Interrupt Vector Table Shown at Right
- Each Entry a Minimum of 32-bits Wide
- Table Entries Can Be Configured to Support Additional Data (larger size)
- First 32-bit Word is a Virtual Address to Start Execution of the Interrupt
- Dummy Interrupt Handlers Should be Used to Populate Unused Interrupts

Vector Table

_CORE_TIMER_VECTOR
_CORE_SOFTWARE_0_VECTOR
_CORE_SOFTWARE_1_VECTOR
_EXTERNAL_0_VECTOR
_TIMER_1_VECTOR
_INPUT_CAPTURE_1_ERROR_VECTOR
_INPUT_CAPTURE_1_VECTOR
_OUTPUT_COMPARE_1_VECTOR
_EXTERNAL_1_VECTOR
_TIMER_2_VECTOR
_INPUT_CAPTURE_2_ERROR_VECTOR
_INPUT_CAPTURE_2_VECTOR
_OUTPUT_COMPARE_2_VECTOR
_EXTERNAL_2_VECTOR
_TIMER_3_VECTOR
_INPUT_CAPTURE_3_ERROR_VECTOR
_INPUT_CAPTURE_3_VECTOR
_OUTPUT_COMPARE_3_VECTOR
_EXTERNAL_3_VECTOR
_TIMER_4_VECTOR
_INPUT_CAPTURE_4_ERROR_VECTOR

Interrupts and Interrupt Vector Tables

- Base Location of Interrupt Vector Table (IVT) Can Be Changed via SFR
 - Facilitates Two Stage Startup with Boot Flash
- Some Systems Allow Placement of IVT in Ram

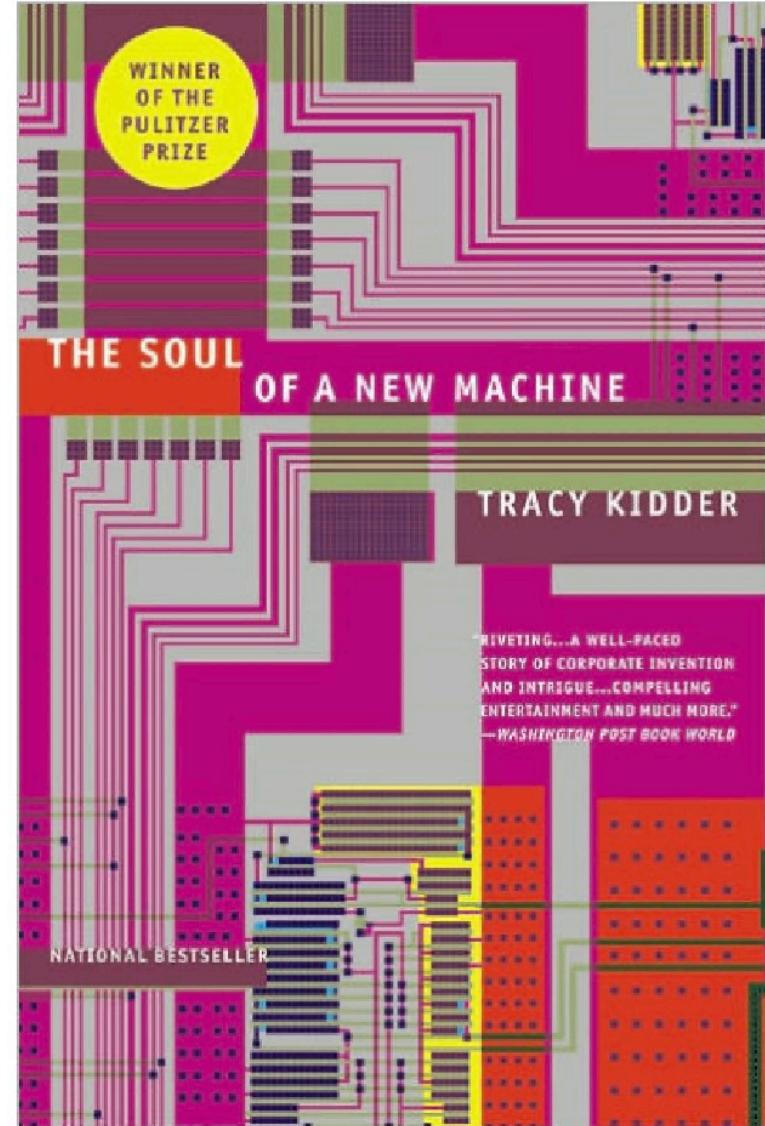


Look Ahead

- Discussion on Reading
- Project Structure
- Preview of Lab 11

Assignment - Readings

- The Soul Of A New Machine
 - One Week Assignment
 - Chapters 5 and 6: Midnight Programmer, Flying Upside Down
 - Send Me Discussion Topics by 10:00 AM on Tuesday, November 12, 2024.





Action Items and Discussion

AI#:	Owner	Slide #	Document	Action