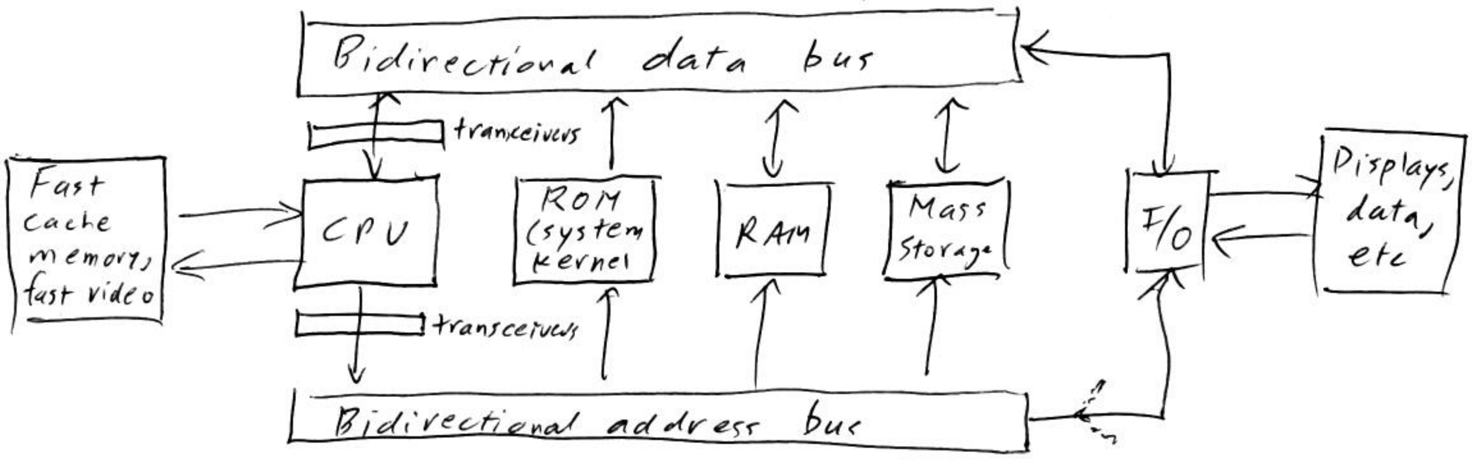
Microprocessors and Microcontrollers

Basic organization of a simple microprocessor



This is typical of an early or small processor such as the 80x86 family, still used for applications like automotive processors.

The CPU has the ability to flexibly change state based on input data (instructions). It has an instruction pointer (address) register and many data registers; may be 8, 16, 32, or 64 bits wide.

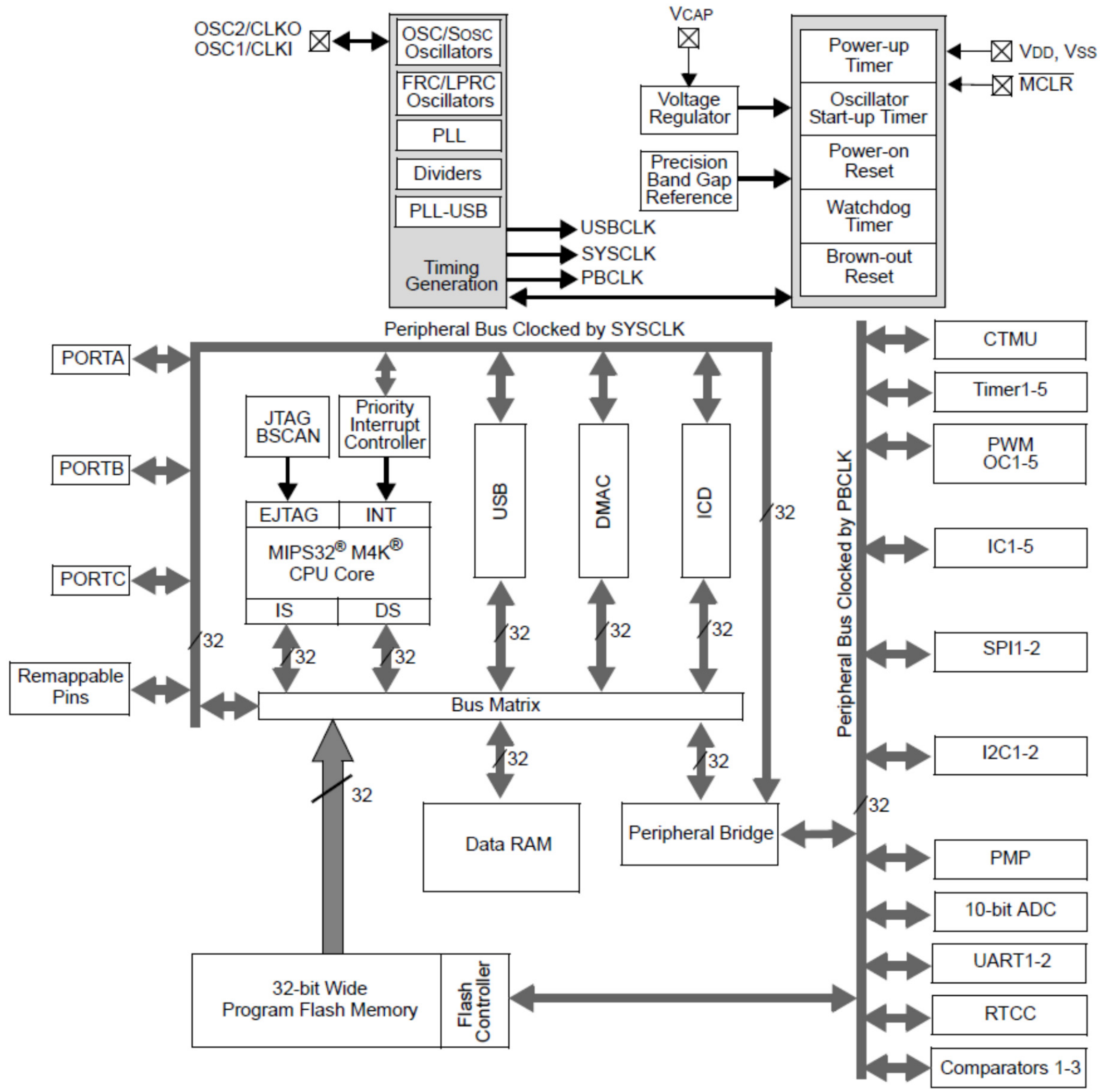
Interrupts are also supported; IP and other registers immediately saved in "stack", execution transfers to high-priority code.

Typically a microprocessor is faster but has no built in peripheral devices or I/O. By contrast, a microcontroller is a self-contained single-chip computer system, in most cases.

On the following pages we take a quick look at the Microchip PIC32MX250 microcontroller (~ $4).

Overall organization of Microchip PIC32MX250 32-bit microcontrollers

FIGURE 1-1: BLOCK DIAGRAM

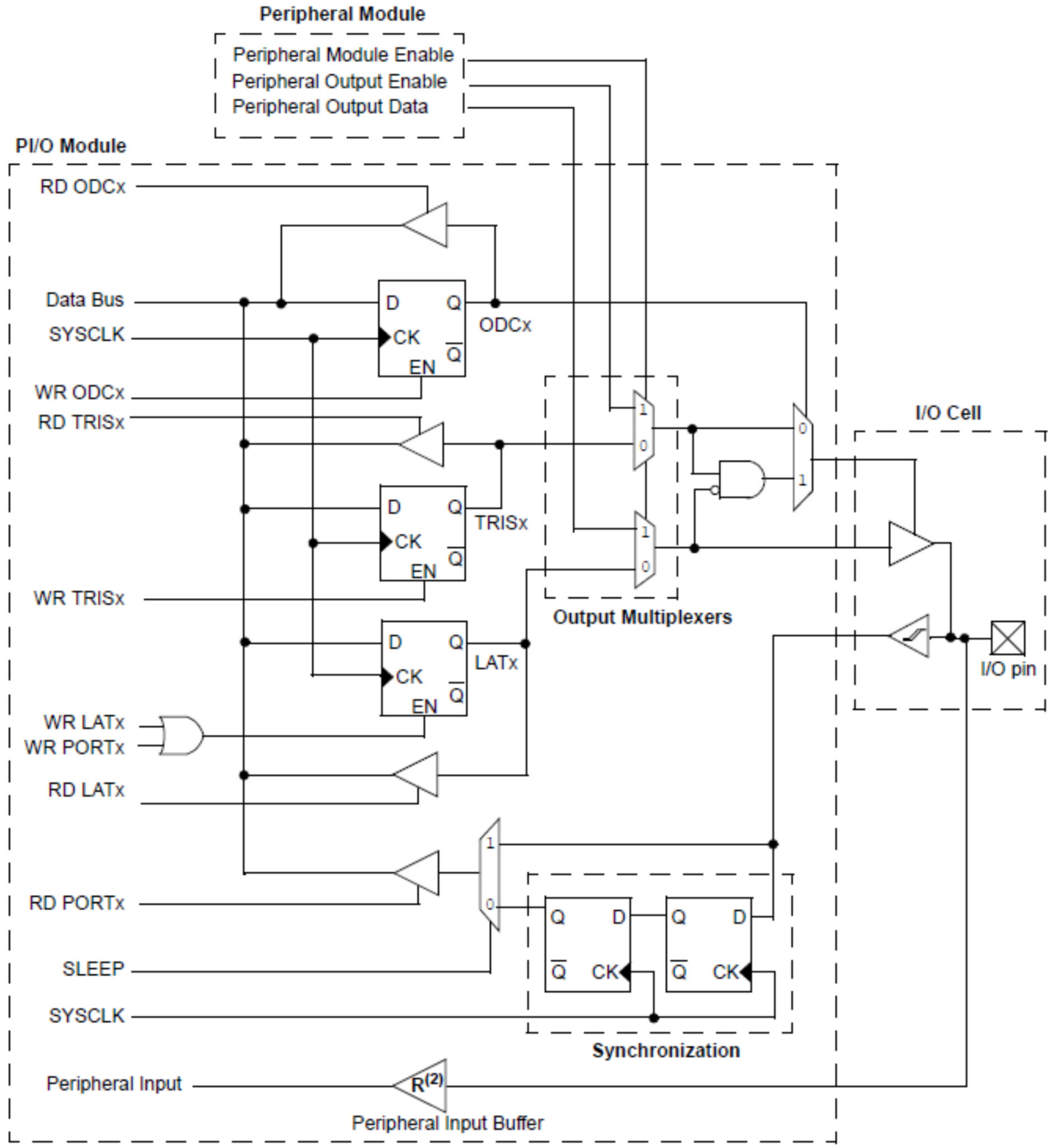


Note: Some features are not available on all devices. Refer to the family features tables (Table 1 and Table 2) for availability.

CTMU: charge/timing measurement unit (for touchpad detection, etc.)
PWM: Pulse-width modulator
SPI, I2C: Clocked serial interfaces
PMP: Parallel master port (a versatile parallel digital data interface)
UART: RS232-type serial interface
RTCC: Real-time clock/calendar
Comparator: Analog comparator
DMAC: Direct memory access controller
ICD, JTAG: Programming and debugging

Organization of I/O pins

Figure 12-4: Block Diagram of a Typical Shared Port Structure(1)

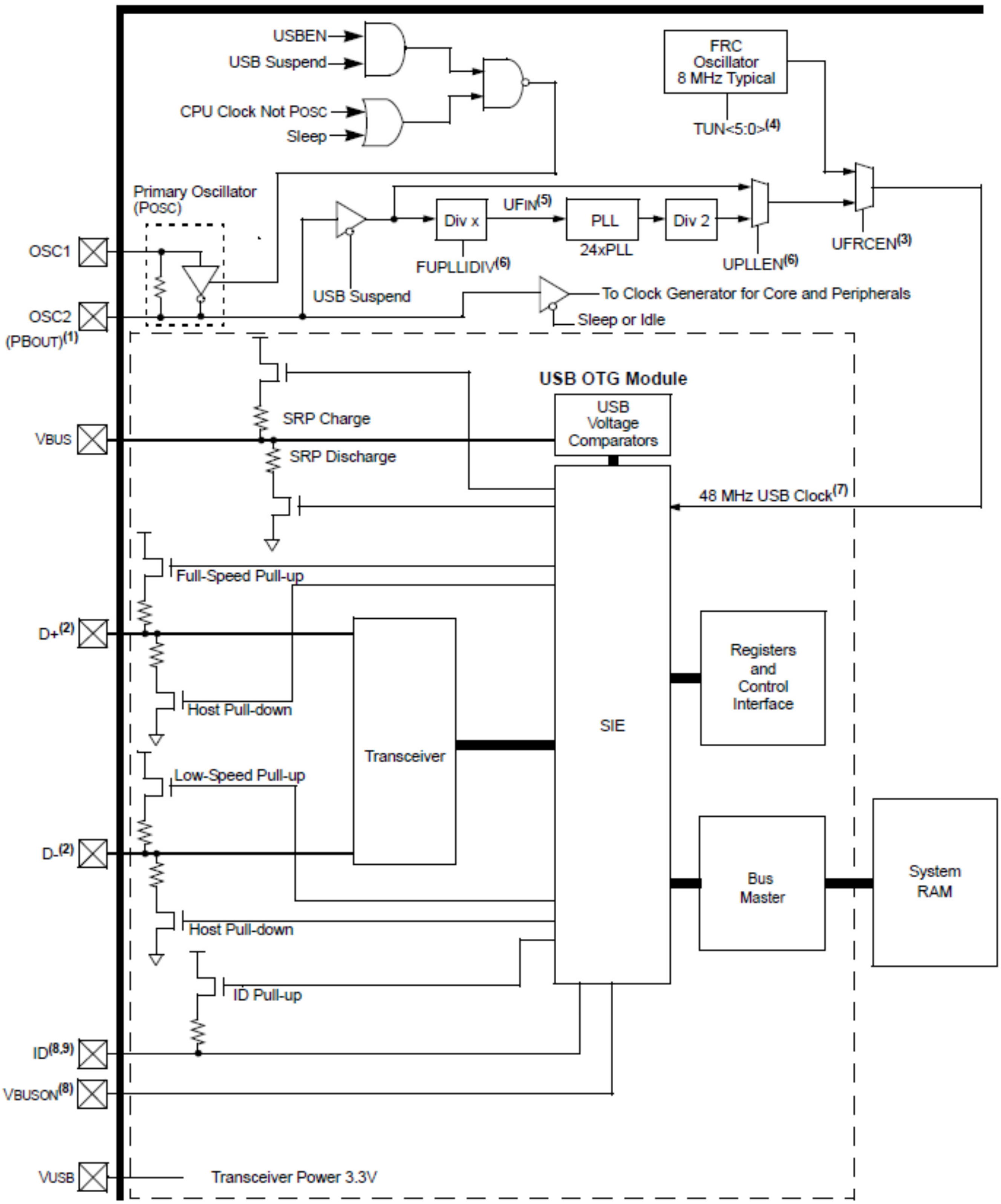


Note 1: This block diagram is a general representation of a shared port/peripheral structure for illustration purposes only. The actual structure for any specific port/peripheral combination may be different than what is shown here.

2: R = Peripheral input buffer types may vary. Refer to the specific device data sheet for peripheral details.

Pins can be used as digital input, digital output, or analog input; when used as outputs they can be configured normally or in open-drain configuration. As inputs, they include optional pullup resistors. Finally, peripheral devices like UARTs and counters can be flexibly assigned to varying pins.

Example of an internal peripheral controller: USB interface unit



Programming

The program memory is written with a special high-voltage programming pulse, generated by a PICkit3 programmer or similar device. The processors have a register and stack structure optimized for programming in C or C++. The actual hardware instruction set consists of primitive operations like the following:

Table 11-7 Instruction Set

Instruction	Description	Function
ADD	Integer Add	Rd = Rs + Rt
ADDI	Integer Add Immediate	Rt = Rs + Immed
ADDIU	Unsigned Integer Add Immediate	Rt = Rs $+_U$ Immed
ADDU	Unsigned Integer Add	Rd = Rs $+_U$ Rt
AND	Logical AND	Rd = Rs & Rt
ANDI	Logical AND Immediate	Rt = Rs & (0_{16} || Immed)
B	Unconditional Branch (Asembler idiom for: BEQ r0, r0, offset)	PC += (int)offset
BAL	Branch and Link (Asembler idiom for: BGEZAL r0, offset)	GPR[31] = PC + 8 PC += (int)offset
BEQ	Branch On Equal	if Rs == Rt PC += (int)offset
BEQL	Branch On Equal Likely	if Rs == Rt PC += (int)offset else Ignore Next Instruction
BGEZ	Branch on Greater Than or Equal To Zero	if !Rs[31] PC += (int)offset
BGEZAL	Branch on Greater Than or Equal To Zero And Link	GPR[31] = PC + 8 if !Rs[31] PC += (int)offset

Most instructions will operate on 32-bit registers in a single instruction cycle; i.e., at a rate of 40-50 MHz. This is equivalent to a microprocessor with a much higher clock speed, since a complete instruction executes on each cycle.

Typical circuit

The schematic diagram below shows a PIC32 microcontroller with a precise crystal clock, a 6-pin programming header, a serial display output, and an SPI interface to a dual 12-bit DAC. It is similar to the configuration used in Lab 10 for Physics 3150.