PIC32 Study Problem Solutions

PIC32 Hardware

1. Hex and binary practice.

a. 0xB15F = 1011 0001 0101 1111 (binary) = 11 x 16^3 + 1 x 16^2 + 5 x 16^1 + 15 x 16^0 = 45407.

b. 207 = 1100 1111 = 0xCF

c. 1010 0010 OR 0100 0110 = 1110 0110 = 0xE6

d. 0010 1101 AND 1110 0011 = 0010 0000 = 0x20

e. 0100 0111 shifted right 3 is 0000 1000 = 0x08

f. 122 (base 10) = 0111 1010 shifted left 2 is 1110 1000 = 0xE8

2. This PIC is the PIC32MX340F128H. It costs between $5.45 and $5.62 (as of Jan 2010) in quantity 1 and comes in QFN packages (Quad Flat No leads; the pads are underneath the chip) and TQFP packages (Thin Quad Flat Pack, with leads coming out to the sides, like the PIC32 on the NU32 board). The price is about $2.00 less than our PIC (as of Jan 2010) in quantity 1, and it has less program memory (128 K vs. 512 K) and lacks USB capability. Other features look similar from the parametric table, but the parametric table does not include all features.

3. Timing generation creates the SYSCLK, PBCLK, and USBCLK signals from an external reference (crystal or resonator) or an internal reference. MCLR is an input that, when held low, resets the PIC. SYSCLK is the main system (CPU) clock, PBCLK is the peripheral bus clock used by many of the peripherals, and USBCLK is a 48 MHz USB-standard clock for USB communication. Ports A-G are I/O ports; each can be used for digital I/O, and port B can be used for analog input. Timer{1..5} are used for counting events, either external signal edges or internal clock pulses. The 10-bit analog-to-digital converter is attached to the 16 inputs of port B and converts an analog voltage to a 10-bit number, 0..1023. Comparators can be used to measure which of two analog inputs has a higher voltage. UARTs are used for asynchronous serial communication, like RS-232 and RS-485. I2C is used for "I squared C" synchronous serial communication. SPI is used for "serial-peripheral interface" synchronous serial communication. IC refers to input capture pins, which can be used to generate interrupts based on external signals. PWM OC are "output compare" peripherals, and our primary use of them is to generate pulse-width modulation signals for motor control. CN are "change notification" peripherals, and these input pins can be used to generate interrupts when the signal changes. Data RAM refers to the 32K memory that can be dynamically written and altered during program execution. Program Flash Memory refers to the 512K memory available to store the program. The Pre-Fetch Module is a cache used to fetch and temporarily store program instructions from (slow) flash memory, to speed up operation when possible.

4. A peripheral refers to capabilities that the PIC has for interfacing with the external world, and include ports for ADC and digital I/O, serial communication, interrupt generation, etc. (including everything on the PBCLK bus).

5. ANx: 10-bit analog inputs. INTx: pins used for external interrupts. OCx: pins used to generate PWM signals. Rxy: The y'th bit of the x'th port, used for digital I/O. SCLx, SDAx: clock and data lines used for I2C communication. SCKx, SDIx, SDOx: clock, data in, and data out lines for SPI communication. SSx: "slave select" lines used for SPI communication to select the specific peripheral to interface with. TxCK: inputs for counting by the timers. UxCTS, UxRTS, UxRX, UxTX: clear to send and request to send (handshaking signals, not always used) and data receive and data transmit lines for asynchronous serial communication like RS-232 and RS-485. VDD: positive power supply for the PIC. VSS: ground for PIC power.

6. C1IN-: comparator negative input. AN4: analog input 4. CN6: change notification pin 6. RB4: digital I/O 4.

7. There are 58 pins on the NU32 board. 52 pins connect directly to the PIC32. 21 of the PIC32's 100 pins are not connected at all. The remaining 27 PIC32 pins are connected internally, on the board.
8. The USB cable provides 5 V power. The PIC uses 3.3 V, provided by a voltage regulator.
9. NU32 LEDs are on when the digital outputs are low.

PIC32 Software

10. The bootloader serves two main purposes: (1) it sets a number of the SFR (Special Function Register) configuration bits specifying how the PIC32 is used on the NU32 board, and (2) it has code that allows the PIC to communicate with a PC via USB, and to program itself with the code sent from the PC, without requiring a special programmer device.
11.

```
#pragma config FPLLMUL  = MUL_20        // PLL Multiplier
#pragma config FPLLIDIV = DIV_2         // PLL Input Divider
#pragma config FPLLODIV = DIV_1         // PLL Output Divider
```

Net result is to mutiply the 8 MHz oscillator frequency by 10.
12. The Hello World procdefs.ld linker file must be different from the one for the bootloader, because the linker file indicates where in program memory the code should be stored. The bootloader code should stay on the PIC always, so any other program (including Hello World) should be stored at a later location in memory.
13. The bits of the SFR of a peripheral indicate how (or if) that peripheral will be used, as most peripherals can operate in a number of different ways.
14. The primary purpose of HardwareProfile_NU32.h is to set four digital I/O pins of the PIC32 to be outputs, to drive the LEDs on the board, and to set two digital I/O pins as inputs, for the USER and PRG switches on the board. TRISE is the "tri-state" register for port E. Setting a bit equal to 1 makes the corresponding pin an input, and setting it to 0 makes it an output. PORTE is the "port" register for port E. Writing to this SFR updates the output values held by digital outputs at pins configured as outputs, and reading from it reads the current values at the pins. LATE is the "latch" SFR for port E. Writing to it sets the outputs of pins set for output. Reading from it reads the "latched" value of pins configured as inputs. (These values may be different from those currently on the pins; they are the values on the pins the last time data was "latched" from the PORT pins.) In this file, menomic reference names are defined for the LED states using bits of LATE, and for the values of the two switches using bits of PORTE. TRISE is used to set 4 pins as ouptuts and 2 pins as inputs.
15. plib.h includes all the peripheral library header files.
16. adc10.h is in Program Files\Microchip\MPLAB C32\pic32-libs\include\peripheral. The number 10 signifies that this is a 10-bit ADC library. (ADCs with different resolution are possible.)
17. p32mx460f512l.h is in Program Files\Microchip\MPLAB C32\pic32-libs\include\proc.
18. The register addresses in the assembly code are separated by 4, because each register is 32 bits, and a memory address specifies the location of a byte, which is 8 bits. So 4 memory locations are needed for each 4-byte SFR.

DONE: this bit indicates whether the A/D conversion has finished.
SAMP: this bit indicates whether the sample-and-hold is currently sampling the input signal, or if it is currently holding a voltage.
ASAM: this bit indicates whether or not sampling begins immediately after the last conversion has completed.
CLRASAM: this bit indicates whether conversions should halt when a specified number of samples have been taken.

SSRC<2:0>: these 3 bits determine whether ADC sampling and conversion is automatic or based on other signals.
FORM<2:0>: these 3 bits determine the format returned by the ADC (e.g., 16-bit integer, 32-bit integer, etc.)
SIDL: this bit determines whether the ADC continues operating when the PIC is in IDLE mode.
FRZ: this bit determines whether the ADC freezes operation when the CPU enters debug exception mode.
ON: this bit determines whether the ADC is on or off.

The POSITION definitions tell us the location of each of these bits in the AD1CON1 SFR. (Note that some of the bits are unused, like bit 3.) The LENGTH definition tell us how many bits are associated with each (1 for each of these examples). The MASK definition means that if we AND the SFR with this mask, the remaining number is all zeros except for the value associated with that bit (which is at bit location POSITION).

FORM: these 3 bits determine the format of the return value. The POSITION is the position of the least signficant of these three bits (bit 8 in the SFR), the LENGTH is 3 (3 bits long), and the MASK with the SFR returns all zeros except for the values of the SFR at bits 8, 9, and 10 (the FORM bits).

The POSITION definitions are used extensively in adc10.h, in the definitions of mnemonic constants. For example, ADC_FORMAT_SIGN_FRACT is defined as the value 3 (binary 011) shifted left 8 times to put the bits in bits 8-10, specifying the format of the ADC return value. MASKs are used a few times in function definitions. For example, #define EnableADC10() (AD1CON1SET = _AD1CON1_ON_MASK). "SET" means that the specified bits are set to 1. ("CLR" means that the specified bits are set to 0.) In this case, we are simply saying that all the bits specified by the mask should be set to 1, which, in this case, turns the ADC peripheral on. The LENGTH definitions are never used in adc10.h.

19. Following the description of the AD1CON1 SFR, we see that in binary, the last 16 bits would be 1010 0100 1110 0110, where bits 3, 11, and 12 are reserved (and written as zeros here). Bits 16-31 are also reserved. Writing these also as zero, we get 0x0000A4E6.
20. mPORTBSetPinsAnalogIn() takes a 32-bit number and uses TRISBSET to "set" (as 1) the TRISB bits that are 1 in the input number (making those pins inputs). It also "clears" those same bits (clears them to 0) in the SFR AD1PCFG, meaning that the pins are set to function as analog inputs, not digital inputs.
21. BusyUART1() and putcUART1() are defined in Program Files\Microchip\MPLAB C32\pic32-libs\include\peripheral\uart.h.