Numair Ahmed
HW3 - ME333, W2021

Chapter 3

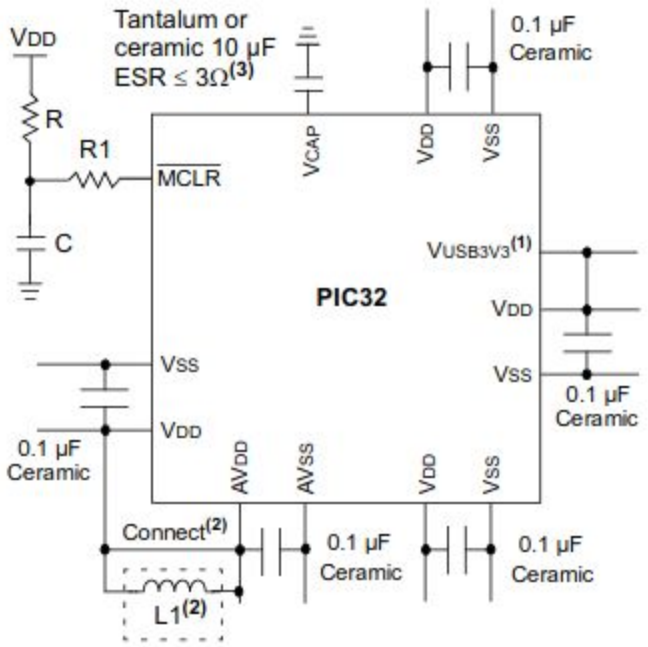
1. a. 0x80000020 + 0x1FFFFFFF = 0xA000001F; cacheable
   b. 0xA0000020 + 0x1FFFFFFF = 0xC000001F; non-cacheable
   c. 0xBF800001 + 0x1FFFFFFF = 0xDF800000; non-cacheable
   d. 0x9FC00111 + 0x1FFFFFFF = 0xBFC00110; cacheable
   e. 0x9D001000 + 0x1FFFFFFF = 0xBD000FFF; cacheable

2. virtual memory location for installing custom programs: _RESET_ADDR = (0xBD000000 + 0x1000 + 0x970);

3. a. the Figure 2.1 in the data sheet doesn't show I/O for PORTB-PORTG. Can't seem to find the info anywhere else in the data sheet either.. Here is what I see for Figure 2.1

PIC32MX5XX/6XX/7XX

FIGURE 2-1: RECOMMENDED MINIMUM CONNECTION



Note 1: If the USB module is not used, this pin must be connected to VDD.

2: As an option, instead of a hard-wired connection, an inductor (L1) can be substituted between VDD and AVDD to improve ADC noise rejection. The inductor impedance should be less than 3Ω and the inductor capacity greater than 10 mA.

Where:

$$f = \frac{F_{CNV}}{2}$$ (i.e., ADC conversion rate/2)

$$f = \frac{1}{(2\pi\sqrt{LC})}$$

$$L = \left(\frac{1}{(2\pi f\sqrt{C})}\right)^2$$

3: Aluminum or electrolytic capacitors should not be used. ESR ≤ 3Ω from -40°C to 125°C @ SYSCLK frequency (i.e., MIPS).

b. pins 30-16 are unimplimented

TABLE 7-3: INTERRUPT REGISTER MAP FOR PIC32MX664F064H, PIC32MX664F128H, PIC32MX675F256H, PIC32MX675F512H AND PIC32MX695F512H DEVICES

Virtual Address (BF88_#)	Register Name(1)	Bit Range	Bits															All Resets
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
1000	INTCON	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SS0	0000
		15:0	—	—	—	MVEC	—	TPC<2:0>			—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP	0000

4. Have not picked up kit yet
5. Have not picked up kit yet

6. a. I2C3CON VA is BF80_5000
   b. TRISC VA is BF88_6080

7. The XC32_bin2hex utility takes an input of .elf formatted binary object files that are linked together and outputs a single .hex file, reducing the size of the file in the process.

8. Don't yet have the PIC32 kit so have not installed the libraries

Chapter 4:

1. All of the register definitions are NU32 specific

```
#pragma config DEBUG = OFF          // Background Debugger disabled
#pragma config FWDTEN = OFF         // WD timer: OFF
#pragma config WDTPS = PS4096       // WD period: 4.096 sec
#pragma config POSCMOD = HS         // Primary Oscillator Mode: High Speed crystal
#pragma config FNOSC = PRIPLL       // Oscillator Selection: Primary oscillator w/ PLL
#pragma config FPLLMUL = MUL_20     // PLL Multiplier: Multiply by 20
#pragma config FPLLIDIV = DIV_2     // PLL Input Divider:  Divide by 2
#pragma config FPLLODIV = DIV_1     // PLL Output Divider: Divide by 1
#pragma config FPBDIV = DIV_1       // Peripheral Bus Clock: Divide by 1
#pragma config UPLLEN = ON          // USB clock uses PLL
#pragma config UPLLIDIV = DIV_2     // Divide 8 MHz input by 2, mult by 12 for 48 MHz
#pragma config FUSBIDIO = ON        // USBID controlled by USB peripheral when it is on
#pragma config FVBUSONIO = ON       // VBUSON controlled by USB peripheral when it is on
#pragma config FSOSCEN = OFF        // Disable second osc to get pins back
#pragma config BWP = ON             // Boot flash write protect: ON
#pragma config ICESEL = ICS_PGx2    // ICE pins configured on PGx2
#pragma config FCANIO = OFF         // Use alternate CAN pins
#pragma config FMIIEN = OFF         // Use RMII (not MII) for ethernet
#pragma config FSRSSEL = PRIORITY_6 // Shadow Register Set for interrupt priority 6

#define NU32_DESIRED_BAUD 230400    // Baudrate for RS232
```

Enabling and setting the prefetch cache, pins, LEDs, and UART is private to the NU32_Startup( ) function.

```
// set the prefectch cache wait state to 2, as per the
// electrical characteristics data sheet
CHECONbits.PFMWS = 0x2;

//enable prefetch for cacheable and noncacheable memory
CHECONbits.PREFEN = 0x3;

// 0 data RAM access wait states
BMXCONbits.BMXWSDRM = 0x0;

// enable multi vector interrupts
INTCONbits.MVEC = 0x1;

// disable JTAG to get B10, B11, B12 and B13 back
DDPCONbits.JTAGEN = 0;

TRISFCLR = 0x0003;  // Make F0 and F1 outputs (LED1 and LED2)
NU32_LED1 = 1;      // LED1 is off
NU32_LED2 = 0;      // LED2 is on

// turn on UART3 without an interrupt
U3MODEbits.BRGH = 0; // set baud to NU32_DESIRED_BAUD
U3BRG = ((NU32_SYS_FREQ / NU32_DESIRED_BAUD) / 16) - 1;

// 8 bit, no parity bit, and 1 stop bit (8N1 setup)
U3MODEbits.PDSEL = 0;
U3MODEbits.STSEL = 0;

// configure TX & RX pins as output & input pins
U3STAbits.UTXEN = 1;
U3STAbits.URXEN = 1;
// configure hardware flow control using RTS and CTS
U3MODEbits.UEN = 2;
```

Reading and writing to UART3 via TX and RX pins is private to NU32.

```
// set the prefectch cache wait state to 2, as per the
// electrical characteristics data sheet
CHECONbits.PFMWS = 0x2;

//enable prefetch for cacheable and noncacheable memory
CHECONbits.PREFEN = 0x3;

// 0 data RAM access wait states
BMXCONbits.BMXWSDRM = 0x0;

// enable multi vector interrupts
INTCONbits.MVEC = 0x1;

// disable JTAG to get B10, B11, B12 and B13 back
DDPCONbits.JTAGEN = 0;

TRISFCLR = 0x0003;  // Make F0 and F1 outputs (LED1 and LED2)
NU32_LED1 = 1;      // LED1 is off
NU32_LED2 = 0;      // LED2 is on

// turn on UART3 without an interrupt
U3MODEbits.BRGH = 0; // set baud to NU32_DESIRED_BAUD
U3BRG = ((NU32_SYS_FREQ / NU32_DESIRED_BAUD) / 16) - 1;

// 8 bit, no parity bit, and 1 stop bit (8N1 setup)
U3MODEbits.PDSEL = 0;
U3MODEbits.STSEL = 0;

// configure TX & RX pins as output & input pins
U3STAbits.UTXEN = 1;
U3STAbits.URXEN = 1;
// configure hardware flow control using RTS and CTS
U3MODEbits.UEN = 2;
```

2. Need to redo after getting kit
4. Need to redo after getting kit