Chapter 3
Instruction Set

INTRODUCTION

This chapter covers the instruction set by tracing each processor block apart. The different addressing modes available on PIC18 processors will be discussed. The role and importance of the STATUS register in the context of arithmetic and logic instructions will be covered as well. In order to introduce the reader to computer architecture, an overview of instruction encoding is covered. Last but not least, the relationship between high level statements, namely C language constructs, and assembly language is explained.

ADDRESSING MODES

Figure 1 shows the processor’s blocks: control unit, ALU, multiplier, I/O ports, and data and program memories along with pertinent hardware. You may refer to it in subsequent sections.

Direct Addressing Mode

Consider the instruction movwf of Table 1. This instruction transfers the contents of the working register WREG to the data memory location specified by operand f. The most significant byte of the instruction (0110111a) is the opcode. Since the 12-bit address does not fit within an instruction word, only the eight least significant bits of the data address (AD7 … AD0) are stored within the instruction. This is shown as ffffffff in the instruction encoding. The 4 most significant bits of the address (AD11 … AD8) must be stored in the so-called Bank Select Register BSR. This direct addressing mode is known as Banked Addressing. Another alternative to this addressing mode is known as Access Bank Addressing. It consists of sign-extend the 8-bit address at run-time in order to form the 12-bit address. Further elaboration on these modes is given subsequently.
Banked Addressing

Since the address bus of data memory is 12-bit wide, the CPU may access up to $2^{12}$ bytes or 4 Kbytes of data memory (address range: 0x000 and 0xFFF). This memory is subdivided into 16 banks (256 bytes each) as shown in Table 2.

When using banked addressing, the bank number (0 … 15) is stored in BSR. This is done via the instruction move literal to BSR as in:

```assembly
movlb 2 ; select bank 2
```