COMPARATIVE ANALYSIS OF PIC16F87X AND PIC18FXX2 MICROCONTROLLERS' MEMORY ORGANIZATION

Nina Bencheva, Volodja Tsonev, Yoana Ruseva

Department of Communication Systems and Technologies, University of Rousse, 8 Studentska Street, 7017 Rousse, Bulgaria, phone: +359 887 746 257, e-mail: nbencheva@ecs.ru.acad.bg

The paper presents a comparative analysis of the memory organization of PIC16F87X and PIC18FXX2 microcontrollers. The differences between the corresponding modules of RAM, EEPROM, FLASH and stack are examined. There are significant improvements of PIC18FXX2 memory organization in the following features: developing of Access Bank which helps to avoid tedious transitions from one bank to another; enhanced indirect addressing of RAM; byte table access to FLASH, ensuring fast transfer of operands from FLASH to RAM; a new operation for erasing of 64 bytes blocks; write performed only in blocks of 8 bytes; Fast Register Stack is available.

Keywords: microcontroller, memory, RAM, FLASH, EEPROM, stack

1. INTRODUCTION

The PIC18FXX2 is intended to make conversions from mid-range controllers PIC16F87X [1] to enhanced controllers as easily as possible. In general the core structure of these two families is saved. Changes to register and bit names are kept to a minimum. Some PIC18FXX2 modules are functionally compatible with the corresponding ones of PIC16F87X and should only require minor changes of the program code due to the differences of the program and data memory maps of the devices. Other modules have considerable differences due to added features, or are absolutely new.

In the paper a comparative analysis of memory organization is presented which may need to be observed when migrating an application from a PIC16F87X device to a PIC18FXX2 device.

2. COMPARATIVE ANALYSIS OF RAM ORGANIZATION

In the PIC16F87X all General Purpose Registers (GPR's) and Special Function Registers (SFR's) are stored in four banks. Each bank can provide up to 128 addresses. Two bits are used to select the bank accessed. The first locations in each bank are reserved for SFR's while the remainder is used for GPR's.

The PIC18FXX2 **[2]** data memory map is divided into 16 banks that contain 256 bytes each. The lower 4 bits of the Bank Select Register (BSR) select which bank will be accessed. The SFRs start at the last location of Bank 15 (0xFFF) and extend downwards. Any remaining space beyond the SFRs in the Bank may be implemented as GPRs. GPRs start at the first location of Bank 0 and grow upwards. The Access Bank is an architectural enhancement which is very useful for code optimization. The Access Bank is comprised of the upper 128 bytes in Bank 15 (SFRs) and the lower

128 bytes in Bank 0. A bit in the instruction word specifies if the operation is to occur in the bank specified by the BSR register or in the Access Bank. This bit is denoted as the 'a' bit. In Fig. 1 the data memory map of PIC18F252/452 is shown.

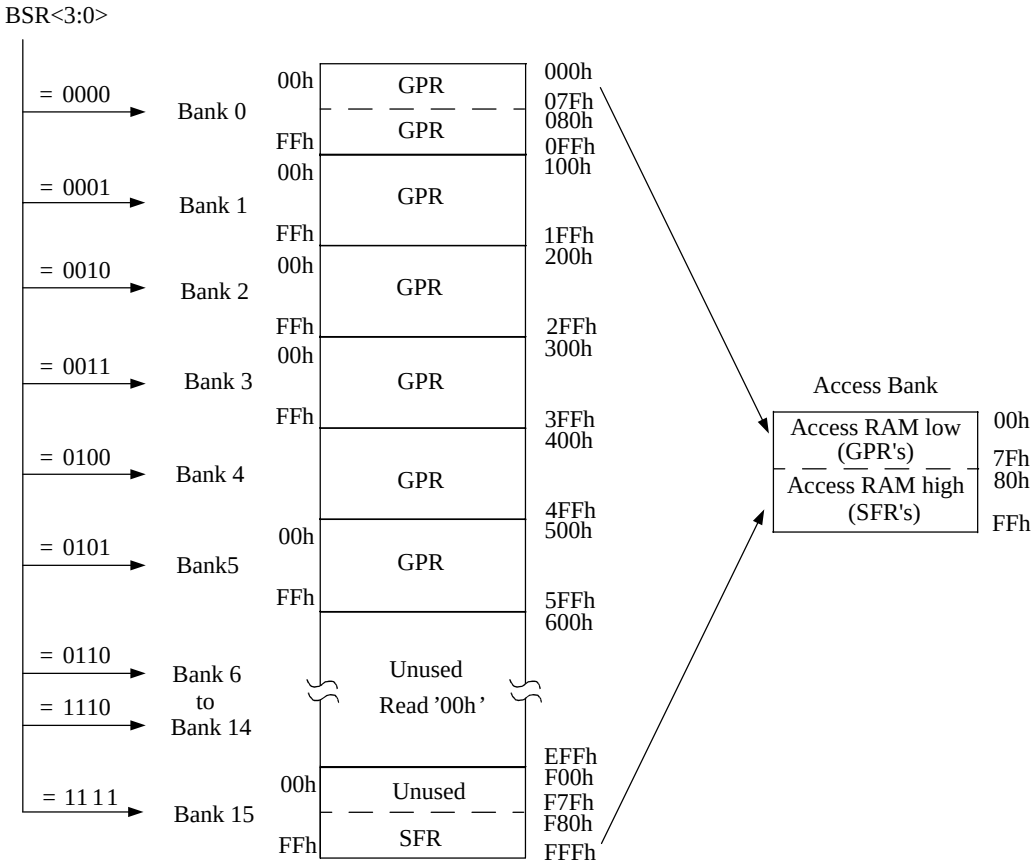

Fig. 1. Data memory map of PIC18F252/452

The registers can be accessed either directly or indirectly through the File Select Register (FSR). There is not any difference in direct addressing for both microcontrollers. In indirect addressing, however, PIC18FXX2 has some additional features. Firstly, there are three indirect addressing registers (pointers) and to address the entire data memory space (4096 bytes), these registers are 12-bit wide. Secondly, there are auto-increment and auto-decrement options associated with indirect addressing as shown in Fig. 2 for FSRi, where i = 0, 1 or 2 [3].

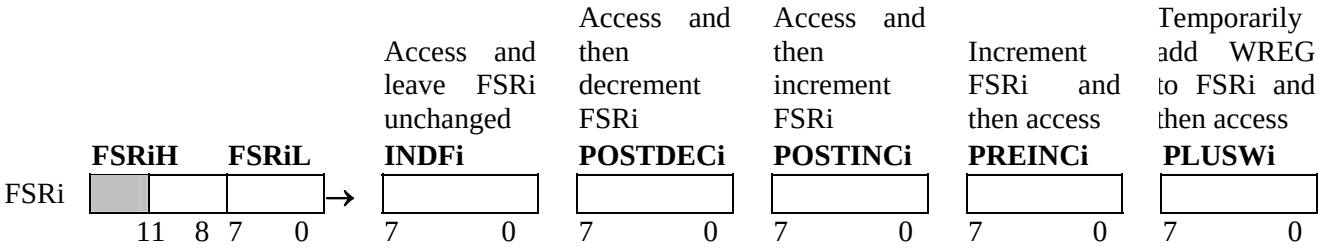

Fig. 2. PIC18FXX2's indirect addressing registers and operand names

3. COMPARATIVE ANALYSIS OF EEPROM ORGANIZATION

The EEPROM data memory of both PIC16F87X and PIC18FXX2 allows byte read and write. The EEPROM data memory is rated for high erase/write cycles. A byte write automatically erases the location and writes the new data (erase-before-write). These devices have 256 bytes of data EEPROM with an address range from 00h to FFh. When interfacing to the EEPROM memory block, EEDATA holds the 8-bit data for read/write and EEADR holds the address of the EEPROM location being accessed. The only difference is the additional FLASH Program/Data EEPROM or Configuration Select bit in EECON1 register of PIC18FXX2.

4. COMPARATIVE ANALYSIS OF FLASH PROGRAM MEMORY ORGANIZATION

The FLASH program memory of PIC16F87X allows word reads and writes. A byte or word write automatically erases the location and writes the new data (erase before write). When interfacing to the program memory block, the EEDATH:EEDATA registers form a 2 byte word which holds the 14-bit data for read/write, and the EEADRH:EEADR registers form a 2 byte word which holds the 13-bit address of the location being accessed. A given sequence for loading EECON2 register must be followed to initiate the write cycle. PIC16F87X can have up to 8K words of FLASH program memory with an address range from 0000h to 3FFFh.

A read from PIC18FXX2 program memory is executed one byte at a time. A write to program memory is executed in blocks of 8 bytes at a time. Program memory is erased in blocks of 64 bytes at a time. The additional FREE bit, implemented in EECON1 register, when set, allows a program memory erase operation that is initiated on the next write command. When FREE is clear, only writes are enabled.

4.1 Table read/write (for PIC18FXX2)

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM – Table Read (TBLRD) and Table Write (TBLWT).

The program memory space is 16-bits wide, while the data RAM space is 8-bits wide. Table Reads and Table Writes move data between these two memory spaces through an 8-bit register TABLAT. The bytes within the program memory are addressed by Table Pointer (TBLPTR). The TBLPTR is comprised of three SFR registers: Table Pointer Upper Byte, Table Pointer High Byte and Table Pointer Low Byte (TBLPTRU:TBLPTRH:TBLPTRL). These three registers form a 22-bit wide pointer. The low order 21 bits allow the device to address up to 2 Mbytes of program memory space. The 22-nd bit allows access to the Device ID, the User ID and the Configuration bits. Fig. 3 shows the operation of a Table Read.

Table Write operations (Fig. 4) store data from the data memory space into holding registers in program memory. A table block containing data is not required to be word aligned. If a Table Write is being used to write an executable code into the program memory, program instructions will need to be word aligned.

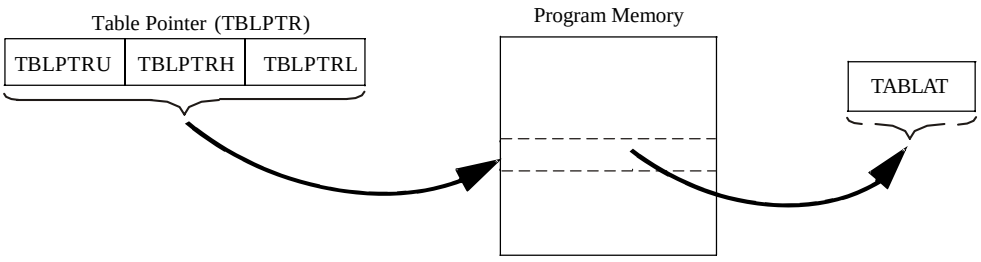

Fig. 3. Table read operation

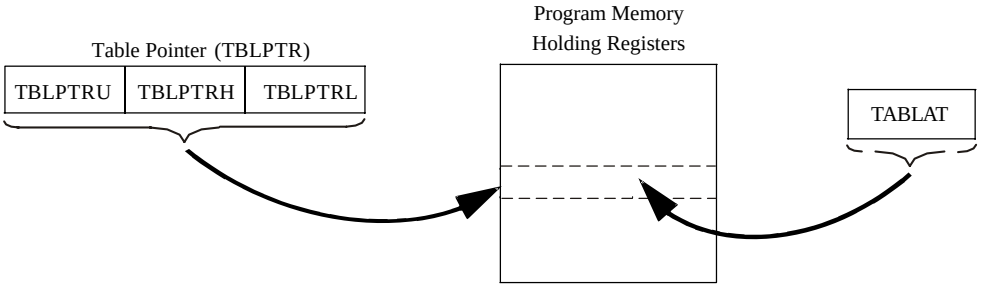

Fig. 4. Table write operation

The TBLRD and TBLWT instructions can update the TBLPTR in one of four ways based on the table operation. These operations are shown in Table 1. They only affect the low order 21 bits.

Table 1

Instruction	Operation on Table Pointer
TBLRD* TBLWT*	TBLPTR is not modified
TBLRD*+ TBLWT*+	TBLPTR is incremented after the read/write
TBLRD*- TBLWT*-	TBLPTR is decremented after the read/write
TBLRD+* TBLWT+*	TBLPTR is incremented before the read/write

4.2 Read/write FLASH program memory (for PIC18FXX2)

Executing TBLRD places the byte pointed by TBLPTR into TABLAT. In addition, TBLPTR can be modified automatically for the next Table Read operation. Fig. 5 shows the interface between the internal program memory and the TABLAT.

Table Writes are used internally to load the holding registers needed to program the FLASH memory. There are 8 holding registers used by the Table Writes for programming. Since the TABLAT is only a single byte, the TBLWT instruction has to be executed 8 times for each programming operation. All of the Table Write operations will essentially be short writes, because only the holding registers are

written. At the end of updating 8 registers, the EECON1 register must be written to, to start the programming operation with a long write. The long write is necessary for programming the FLASH. Fig. 6 shows table writes to the FLASH memory. As in the case of PIC16F87X a given sequence for loading EECON2 register must be followed to initiate the write cycle.

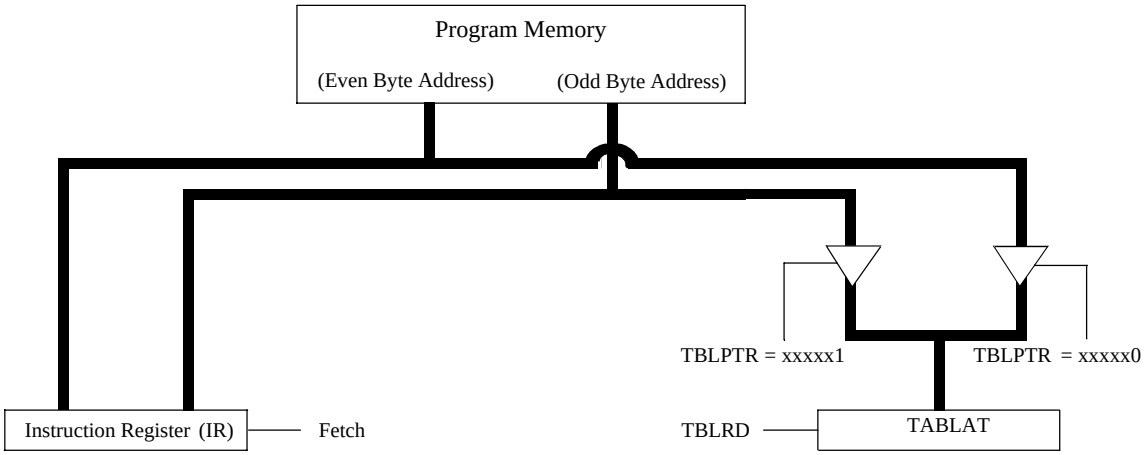

Fig. 5. Read from Flash program memory

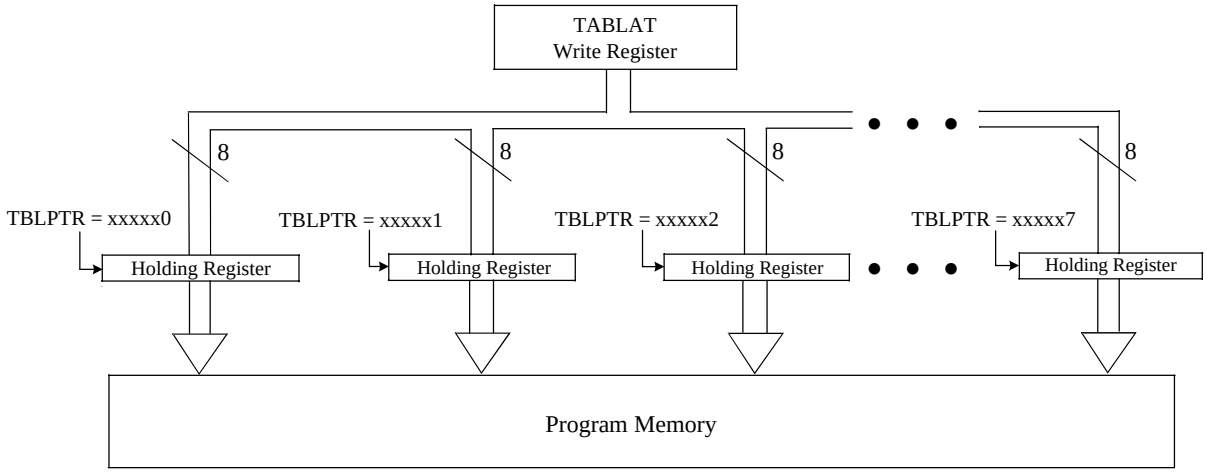

Fig. 6. Table writes to the FLASH program memory

5. COMPARATIVE ANALYSIS OF STACK MEMORY ORGANIZATION

The midrange devices, such as PIC16F87X, have an 8 level deep x 13-bit wide hardware stack. It allows a combination of up to 8 program calls and interruptions to occur. The stack contains the return address from these kinds of branches in program execution. The stack space is not part of either program or data space. The stack pointer is not readable or writable.

The stack of PIC18FXX2 operates as a 31-word by 21-bit RAM with a 5-bit stack pointer (STKPTR) initialized to 00000b after all RESETs. During an instruction, causing a push onto the stack, the stack pointer is first incremented and the RAM

location pointed to by the STKPTR is written with the contents of the program counter (PC). During an instruction, causing a pop from the stack, the contents of the RAM location pointed to by the STKPTR are transferred to the PC and then the stack pointer is decremented. The stack space is not part of either program or data space. The stack pointer is readable and writable, and the address on the top of the stack is readable and writable through SFR registers. Data can also be pushed to, or popped from the stack using the top-of-stack SFRs. Status bits indicate if the stack pointer is at or beyond the 31 levels provided. The top of the stack is readable and writable. Three registers, TOSU, TOSH and TOSL, hold the contents of the stack location pointed to by the STKPTR register. This allows users to implement a software stack if necessary.

A Fast Register Stack is provided for the STATUS, WREG and BSR registers. It has only one level. This stack is not readable or writable and is loaded with the current value of the corresponding register when the processor vectors for an interrupt. The values of the registers are loaded back, if the FAST RETURN instruction is used to return from the interruption. If the interruptions are disabled, the fast register stack can be used to restore the STATUS, WREG and BSR registers at the end of a subroutine call.

6. CONCLUSIONS

The page organization of PIC16F87X RAM is retained and expanded with PIC18FXX2 microcontroller. Developing of Access Bank helps to avoid tedious transitions from one bank to another. There is a significant improvement of indirect addressing of RAM.

The EEPROM organization and read/write in this memory is the same for both families.

There is a significant difference in the organization of PIC18FXX2 FLASH program memory access. Byte table access to FLASH is provided, which ensures fast transfer of operands from FLASH to RAM. A new operation for erasing of blocks of 64 bytes is implemented. Write can be performed only in blocks of 8 bytes.

The stack memory of PIC18FXX2 is user accessible within its capacity determined. The Fast Register Stack provided can be used to restore the STATUS, Work and Bank Select registers at the end of a subroutine call.

The PIC18FXX2 memory features are significantly improved compared to those of PIC16F87X. This enlarges the fields of PIC18FXX2 application.

7. REFERENCES

[1] Microchip Technology Inc., *PICmicro™ Mid-Range MCU Family Reference Manual.* DS30292A, 1998.

[2] Microchip Technology Inc., *PIC18FXX2 Data Sheet,* DS39564A, 2001.

[3] Peatman J. B., *Embedded Design with the PIC18F452 Microcontroller*, Pearson Education Inc, ISBN 0-13-046213-6, 2003.