8.5 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt e.g. W register and STATUS register. This has to be implemented in software.

The action of saving information is commonly referred to as “PUSHing,” while the action of restoring the information before the return is commonly referred to as “POPing.” These (PUSH, POP) are not instruction mnemonics, but are conceptual actions. This action can be implemented by a sequence of instructions. For ease of code transportability, these code segments can be made into MACROs (see MPASM Assembler User’s Guide for details on creating macros).

Example 8-1 stores and restores the STATUS and W registers for devices with common RAM (such as the PIC16C77). The user register, W_TEMP, must be defined across all banks and must be defined at the same offset from the bank base address (i.e., W_TEMP is defined at 0x70 - 0x7F in Bank0). The user register, STATUS_TEMP, must be defined in Bank0, in this example STATUS_TEMP is also in Bank0.

The steps of Example 8-1:

1. Stores the W register regardless of current bank.
2. Stores the STATUS register in Bank0.
3. Executes the Interrupt Service Routine (ISR) code.
4. Restores the STATUS (and bank select bit register).
5. Restores the W register.

If additional locations need to be saved before executing the Interrupt Service Routine (ISR) code, they should be saved after the STATUS register is saved (step 2), and restored before the STATUS register is restored (step 4).

Example 8-1: Saving the STATUS and W Registers in RAM (for Devices with Common RAM)

```
MOVWF   W_TEMP          ; Copy W to a Temporary Register
                        ;    regardless of current bank
SWAPF   STATUS,W        ; Swap STATUS nibbles and place
                        ;    into W register
MOVWF   STATUS_TEMP     ; Save STATUS to a Temporary register
                        ;    in Bank0
:
: (Interrupt Service Routine (ISR) )
:
SWAPF   STATUS_TEMP,W   ; Swap original STATUS register value
                        ;    into W (restores original bank)
MOVWF   STATUS          ; Restore STATUS register from
                        ;    W register
SWAPF   W_TEMP,F        ; Swap W_Temp nibbles and return
                        ;    value to W_Temp
SWAPF   W_TEMP,W        ; Swap W_Temp to W to restore original
                        ;    W value without affecting STATUS
```

© 1997 Microchip Technology Inc.

Example 8-2 stores and restores the STATUS and W registers for devices without common RAM (such as the PIC16C74A). The user register, W_TEMP, must be defined across all banks and must be defined at the same offset from the bank base address (i.e., W_TEMP is defined at 0x70 - 0x7F in Bank0). The user register, STATUS_TEMP, must be defined in Bank0.

Within the 70h - 7Fh range (Bank0), wherever W_TEMP is expected the corresponding locations in the other banks should be dedicated for the possible saving of the W register.

The steps of Example 8-2:

1. Stores the W register regardless of current bank.
2. Stores the STATUS register in Bank0.
3. Executes the Interrupt Service Routine (ISR) code.
4. Restores the STATUS (and bank select bit register).
5. Restores the W register.

If additional locations need to be saved before executing the Interrupt Service Routine (ISR) code, they should be saved after the STATUS register is saved (step 2), and restored before the STATUS register is restored (step 4).

Example 8-2: Saving the STATUS and W Registers in RAM (for Devices without Common RAM)

```
MOVWF   W_TEMP         ; Copy W to a Temporary Register
                       ;    regardless of current bank
SWAPF   STATUS,W       ; Swap STATUS nibbles and place
                       ;    into W register
BCF     STATUS,RP0     ; Change to Bank0 regardless of
                       ;    current bank
MOVWF   STATUS_TEMP    ; Save STATUS to a Temporary register
                       ;    in Bank0
:
: (Interrupt Service Routine (ISR) )
:
SWAPF   STATUS_TEMP,W  ; Swap original STATUS register value
                       ;    into W (restores original bank)
MOVWF   STATUS         ; Restore STATUS register from
                       ;    W register
SWAPF   W_TEMP,F       ; Swap W_Temp nibbles and return
                       ;    value to W_Temp
SWAPF   W_TEMP,W       ; Swap W_Temp to W to restore original
                       ;    W value without affecting STATUS
```

Example 8-3 stores and restores the STATUS and W registers for devices with general purpose RAM only in Bank0 (such as the PIC16C620). The Bank must be tested before saving any of the user registers. , W_TEMP, must be defined across all banks and must be defined at the same offset from the bank base address. The user register, STATUS_TEMP, must be defined in Bank0.

The steps of Example 8-3:

1. Test current bank.
2. Stores the W register regardless of current bank.
3. Stores the STATUS register in Bank0.
4. Executes the Interrupt Service Routine (ISR) code.
5. Restores the STATUS (and bank select bit register).
6. Restores the W register.

If additional locations need to be saved before executing the Interrupt Service Routine (ISR) code, they should be saved after the STATUS register is saved (step 2), and restored before the STATUS register is restored (step 4).

Example 8-3: Saving the STATUS and W Registers in RAM (for Devices with General Purpose RAM Only in Bank0)

```
Push
      BTFSS   STATUS, RP0         ; In Bank 0?
      GOTO    RP0CLEAR            ; YES,
      BCF     STATUS, RP0         ; NO, Force to Bank 0
      MOVWF   W_TEMP              ; Store W register
      SWAPF   STATUS, W           ; Swap STATUS register and
      MOVWF   STATUS_TEMP         ;   store in STATUS_TEMP
      BSF     STATUS_TEMP, 1      ; Set the bit that corresponds to RP0
      GOTO    ISR_Code            ; Push completed
RP0CLEAR
      MOVWF   W_TEMP              ; Store W register
      SWAPF   STATUS, W           ; Swap STATUS register and
      MOVWF   STATUS_TEMP         ;   store in STATUS_TEMP
;
ISR_Code
      :
      : (Interrupt Service Routine (ISR) )
      :
;
Pop
      SWAPF   STATUS_TEMP, W      ; Restore Status register
      MOVWF   STATUS              ;
      BTFSS   STATUS, RP0         ; In Bank 1?
      GOTO    Restore_WREG        ; NO,
      BCF     STATUS, RP0         ; YES, Force Bank 0
      SWAPF   W_TEMP, F           ; Restore W register
      SWAPF   W_TEMP, W           ;
      BSF     STATUS, RP0         ; Back to Bank 1
      RETFIE                      ; POP completed
Restore_WREG
      SWAPF   W_TEMP, F           ; Restore W register
      SWAPF   W_TEMP, W           ;
      RETFIE                      ; POP completed
```