3.4.6.35 PUSH

PUSH[.W] Push word onto stack

PUSH.B Push byte onto stack

Syntax

```
PUSH src  or  PUSH.W src
PUSH.B src
```

Operation SP - 2 → SP

src → @SP

Description The stack pointer is decremented by two, then the source operand is moved to the RAM word addressed by the stack pointer (TOS).

Status Bits Status bits are not affected.

Mode Bits OSCOFF, CPUOFF, and GIE are not affected.

Example The contents of the status register and R8 are saved on the stack.

```
PUSH     SR      ; save status register
PUSH     R8      ; save R8
```

Example The contents of the peripheral TCDAT is saved on the stack.

```
PUSH.B   &TCDAT  ; save data from 8-bit peripheral module,
                 ; address TCDAT, onto stack
```

NOTE: System Stack Pointer

The System stack pointer (SP) is always decremented by two, independent of the byte suffix.

3.4.6.36 RET

***RET** Return from subroutine

Syntax `RET`

Operation @SP → PC

SP + 2 → SP

Emulation `MOV @SP+,PC`

Description The return address pushed onto the stack by a CALL instruction is moved to the program counter. The program continues at the code address following the subroutine call.

Status Bits Status bits are not affected.

3.4.6.37 RETI

RETI Return from interrupt

Syntax `RETI`

Operation TOS → SR

SP + 2 → SP

TOS → PC

SP + 2 → SP

Description The status register is restored to the value at the beginning of the interrupt service routine by replacing the present SR contents with the TOS contents. The stack pointer (SP) is incremented by two.

The program counter is restored to the value at the beginning of interrupt service. This is the consecutive step after the interrupted program flow. Restoration is performed by replacing the present PC contents with the TOS memory contents. The stack pointer (SP) is incremented.

Status Bits N: Restored from system stack

Z: Restored from system stack

C: Restored from system stack

V: Restored from system stack

Mode Bits OSCOFF, CPUOFF, and GIE are restored from system stack.

Example Figure 3-14 illustrates the main program interrupt.

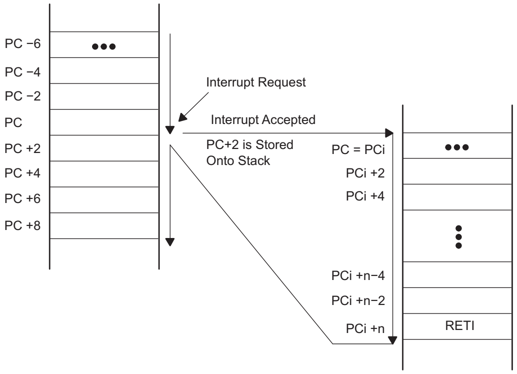

Figure 3-14. Main Program Interrupt