

§ 8. Întreruperi

- Sunt trei tipuri de întreruperi:
 - reset;
 - mascabile;
 - nemascabile.
- Întreruperile se diferențiază după prioritatea lor. Cea mai mare prioritate o are întreruperea *Reset*, care corespunde semnalului */Reset*. Este urmată de întreruperea nemascabilă (*NMI*), care corespunde semnalului *NMI*. Cea mai scăzută prioritate o au întreruperile 4-15, care corespund semnalelor INT4-INT15. Prioritățile întreruperilor sunt prezentate în tabelul următor.

Prioritățile întreruperilor

Priority	Interrupt Name	Interrupt Type
Highest	Reset	Reset
	NMI	Nonmaskable
	INT4	Maskable
	INT5	Maskable
	INT6	Maskable
	INT7	Maskable
	INT8	Maskable
	INT9	Maskable
	INT10	Maskable
	INT11	Maskable
	INT12	Maskable
	INT13	Maskable
	INT14	Maskable
	Lowest	INT15

□ Considerații privind întreruperea RESET

- Întreruperea are loc când semnalul /RESET este activ low. Toate celelalte întreruperi au loc când semnalele sunt active high.
- Pentru ca întreruperea să aibă loc este necesar ca semnalul /RESET să fie activ low minimum 10 cicluri de tact ale procesorului.
- La apariția acestei întreruperi execuțiile instrucțiunilor în curs sunt oprite, iar conținutul registrelor va fi cel implicit.
- Pachetul extras corespunzător serviciului întreruperii trebuie să fie amplasat la adresa specifică.
- Întreruperea nu este afectată de instrucțiuni de salt.

□ Considerații privind întreruperea nemascabilă NMI

- Este, în general utilizată, pentru a alerta CPU de probleme hardware, cum ar fi de exemplu iminenta cădere a puterii.
- Pentru ca întreruperea NMI să aibă loc este necesar ca **bitul NMIE** din **registru de validare a întreruperilor IER** (Interrupt Enable Register) să fie setat pe 1L. Singura condiție care împiedică ca întreruperea NMI să aibă loc când NMIE = 1L este ca această să aibă loc pe durata ciclurilor adiționale (delay slots) ale unei instrucțiuni de salt.
- NMIE este setat pe 0L la inițializarea procesorului pentru a preveni apariția, în acest caz, a unei întreruperi NMI. De asemenea, este setat pe 0L la apariția unei întreruperi NMI pentru a preveni apariția unei alte întreruperi NMI. Cât timp bitul NMIE are valoarea 0L întreruperile mascabile INT4 – INT15 sunt invalidate.
- Bitul NMIE se poate seta prin programare numai pe 0L.

□ Considerații privind întreruperile mascabile INT4 – INT15

- Aceste întreruperi pot fi asociate circuitelor externe, perifericelor din cadrul procesorului sau prin control software.
- Presupunând că întreruperea mascabilă nu apare pe durata ciclurilor adiționale (delay slots) ale unei instrucțiuni de salt, următoarele condiții trebuie îndeplinite pentru a avea loc o întrerupere mascabilă:
 - **bitul de validare a întreruperilor globale GIE** (Global Interrupt Enable) din **regitrul de control stare CSR** (Control Status Register) este setat pe 1L;
 - bitul NMIE este setat pe 1L;
 - **bitul de validare a întreruperilor IE** (Interrupt Enable) din registrul IER este setat pe 1L;
 - când întreruperea corespunzătoare apare, bitul corespunzător din registrul IFR este setat pe 1L și, astfel, nu există biți indicatori de întrerupere cu prioritate ridicată IF (Interrupt Flag) în registrul IFR.

□ Tabelul pentru tratarea întreruperilor IST (Interrupt Service Table)

- Când CPU începe procesarea unei întreruperi, ea se referă la tabelul pentru tratarea întreruperilor IST.
- IST este un tabel de pachete extrase care conțin codurile pentru tratarea întreruperilor.
- IST conține 16 pachete extrase consecutive. Fiecare pachet extras pentru tratarea întreruperii (**ISFP**) conține până la 14 instrucțiuni. O subrutină simplă de tratare a întreruperii trebuie să se încadreze în pachetul extras.
- Adresele și conținuturile tabelului IST sunt prezentate în continuare.

Tabelul IST

xxxx 000h	RESET ISFP
xxxx 020h	NMI ISFP
xxxx 040h	Reserved
xxxx 060h	Reserved
xxxx 080h	INT4 ISFP
xxxx 0A0h	INT5 ISFP
xxxx 0C0h	INT6 ISFP
xxxx 0E0h	INT7 ISFP
xxxx 100h	INT8 ISFP
xxxx 120h	INT9 ISFP
xxxx 140h	INT10 ISFP
xxxx 160h	INT11 ISFP
xxxx 180h	INT12 ISFP
xxxx 1A0h	INT13 ISFP
xxxx 1C0h	INT14 ISFP
xxxx 1E0h	INT15 ISFP

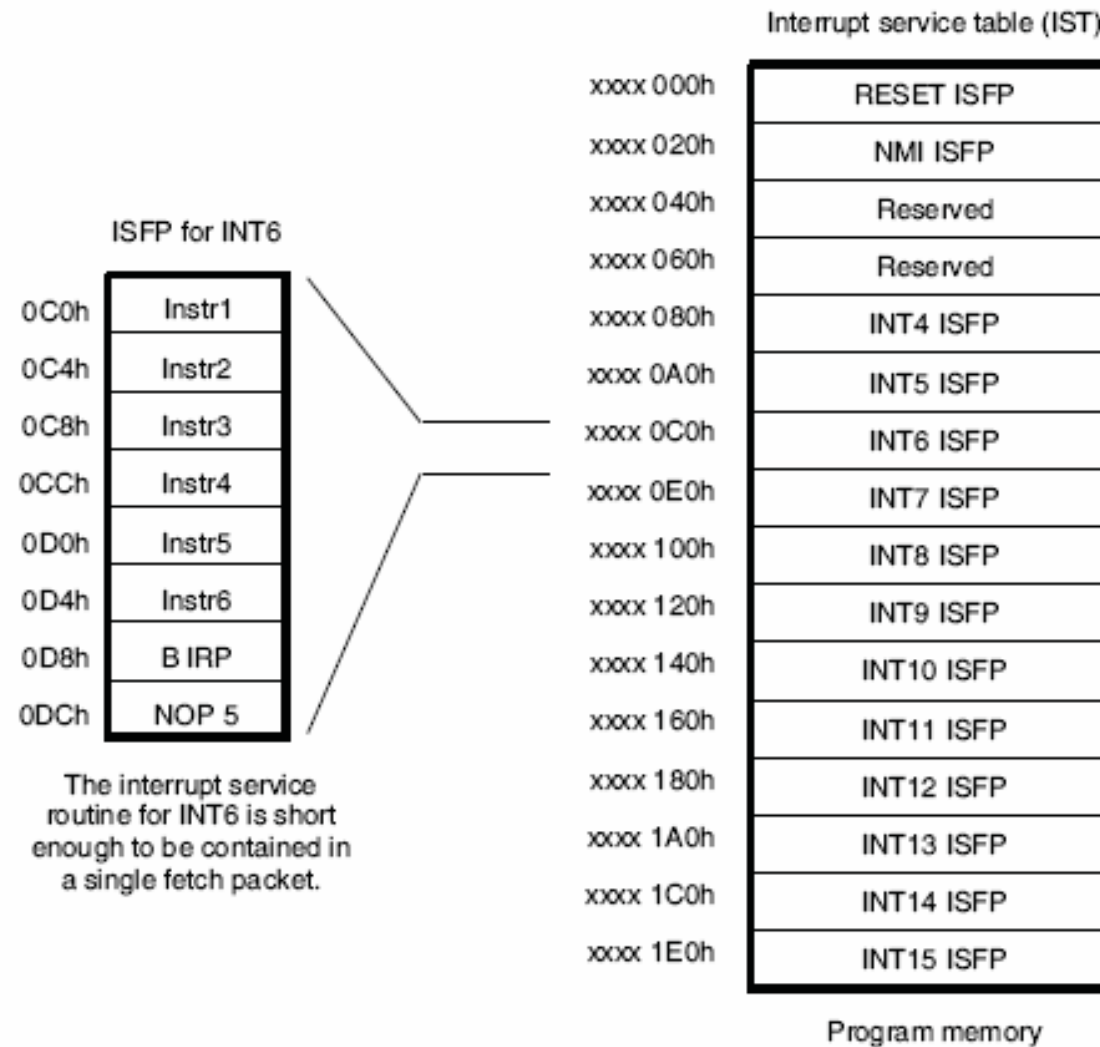
Program memory

Deoarece fiecare pachet extras conține 8 instrucțiuni de 32 de biți (32 octeți), fiecare adresă din tabel este incrementată cu 20h față de precedenta.

□ Pachetul extras pentru tratarea întreruperilor ISFP (Interrupt Service Fetch Packet)

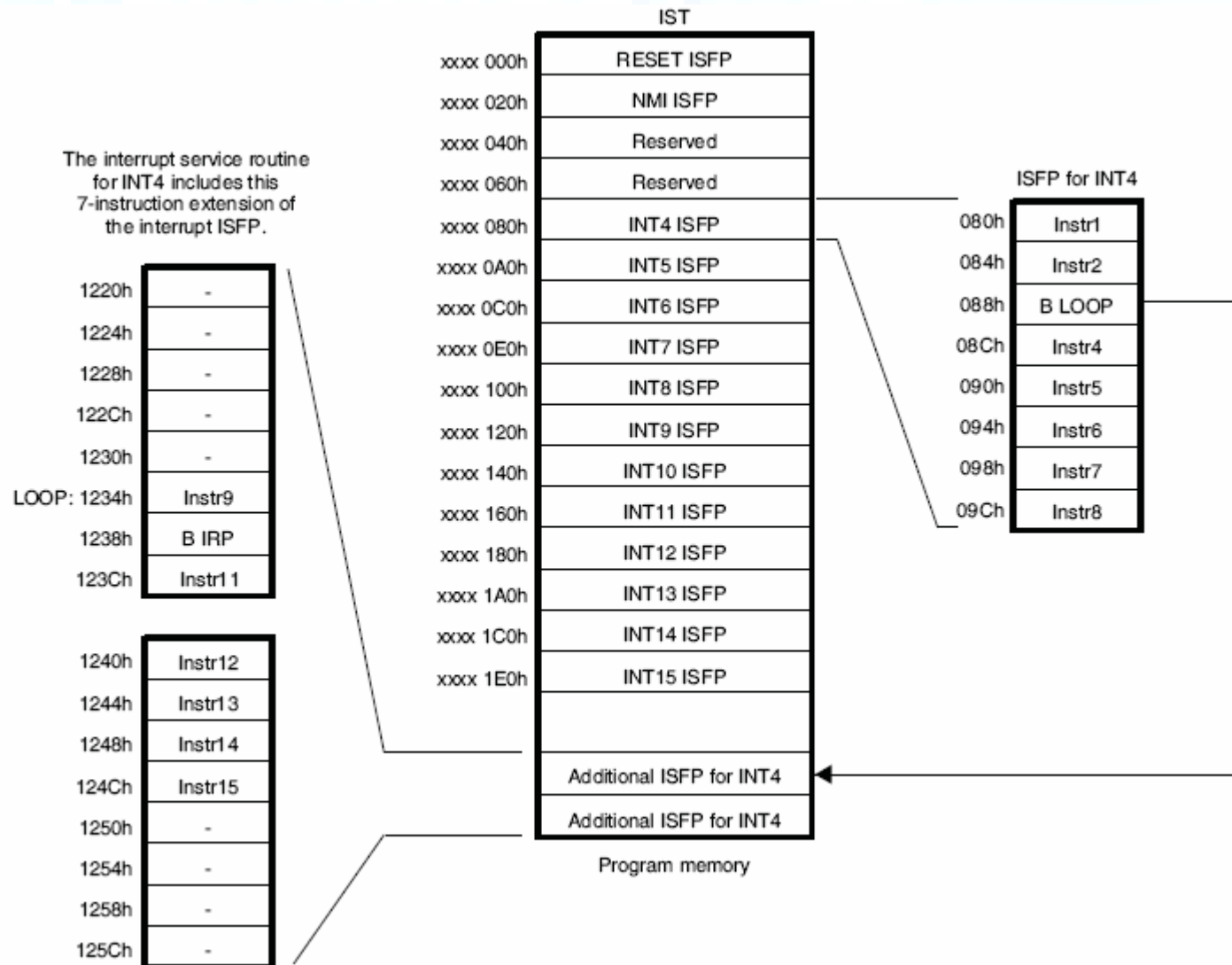
- ISFP este un pachet extras utilizat pentru tratarea unei întreruperi.
- În figura următoare se prezintă pachetul ISFP care conține o subrutină de tratare a întreruperii suficient de redusă pentru a intra într-un singur pachet extras (FP).
- Instrucțiunea **B IRP** realizează revenirea în programul principal – este instrucțiune de revenire la pointerul instrucțiunii (Interrupt Return Pointer).
- Instrucțiunea **NOP 5** permite saltul aferent instrucțiunii **B IRP** să ajungă în faza de pipeline.

Subrutină de tratare a întreruperii care intra într-un pachet extras



- În cazul în care subrutina de tratare a întreruperii nu intră într-un pachet extras, atunci este necesar saltul în locația unei subrutine adiționale de tratare a întreruperii.
- Pentru a completa rutina de tratare a întreruperii are loc un salt în locația de memorie cu adresa 1234h.

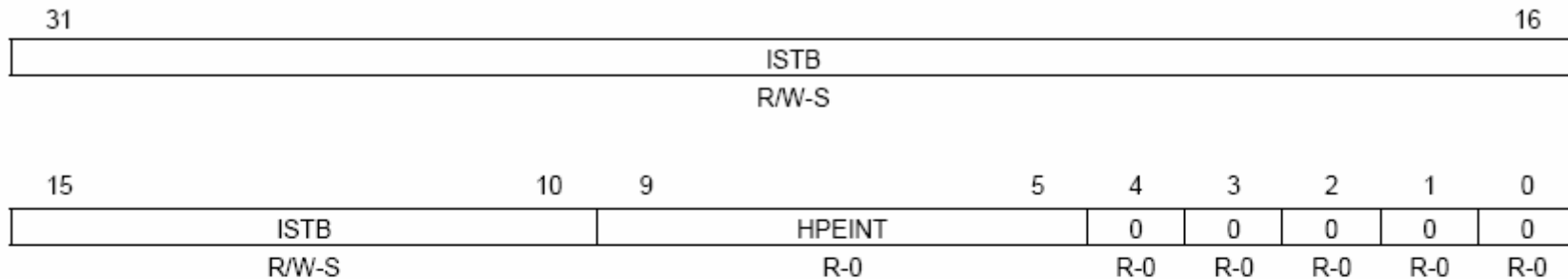
Subrutină de tratare a întreruperii care nu intra într-un pachet extras



□ Pointerul către tabelul subrutinei de tratare a întreruperii

- Pachetul de extragere pentru Reset trebuie să fie stabilit în locația implicită. În schimb, pentru restul întreruperilor IST poate fi în orice locație a memoriei de program (într-o zonă 256 cuvinte sau 1K octet).
- Locația lui IST este determinată de către câmpul bază a tabelului subrutinei de tratare a întreruperii **ISTB** (Interrupt Service Table Basse) a registrului pointer la tabelul subrutinei de tratare a întreruperii **ISTP** (Interrupt Service Table Pointer).

Figure 2-12. Interrupt Service Table Pointer Register (ISTP)



LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -n = value after reset; S = See the device-specific data manual for the default value of this field after reset

Table 2-15. Interrupt Service Table Pointer Register (ISTP) Field Descriptions

Bit	Field	Value	Description
31-10	ISTB	0-3F FFFFh	Interrupt service table base portion of the IST address. This field is cleared to a device-specific default value on reset; therefore, upon startup the IST must reside at this specific address. See the device-specific data manual for more information. After reset, you can relocate the IST by writing a new value to ISTB. If relocated, the first ISFP (corresponding to RESET) is never executed via interrupt processing, because reset clears the ISTB to its default value. See Example 5-1 .
9-5	HPEINT	0-1Fh	Highest priority enabled interrupt that is currently pending. This field indicates the number (related bit position in the IFR) of the highest priority interrupt (as defined in Table 5-1) that is enabled by its bit in the IER. Thus, the ISTP can be used for manual branches to the highest priority enabled interrupt. If no interrupt is pending and enabled, HPEINT contains the value 0. The corresponding interrupt need not be enabled by NMIE (unless it is NMI) or by GIE.
4-0	0	0	Cleared to 0 (fetch packets must be aligned on 8-word (32-byte) boundaries).

Realocare IST

(a) Relocating the IST to 800h

1) Copy IST, located between 0h and 200h, to the memory location between 800h and A00h.

2) Write 800h to ISTOP: `MVK 800h, A2`
`MVC A2, ISTOP`

ISTP = 800h = 1000 0000 0000b

(b) How the ISTOP directs the CPU to the appropriate ISFP in the relocated IST

Assume: IFR = BBC0h = 1011 1011 1100 0000b
 IER = 1230h = 0001 0010 0011 0001b

2 enabled interrupts pending: INT9 and INT12

The 1s in IFR indicate pending interrupts; the 1s in IER indicate the interrupts that are enabled. INT9 has a higher priority than INT12, so HPEINT is encoded with the value for INT9, 01001b.

HPEINT corresponds to bits 9-5 of the ISTOP:
 ISTOP = 1001 0010 0000b = 920h = address of INT9

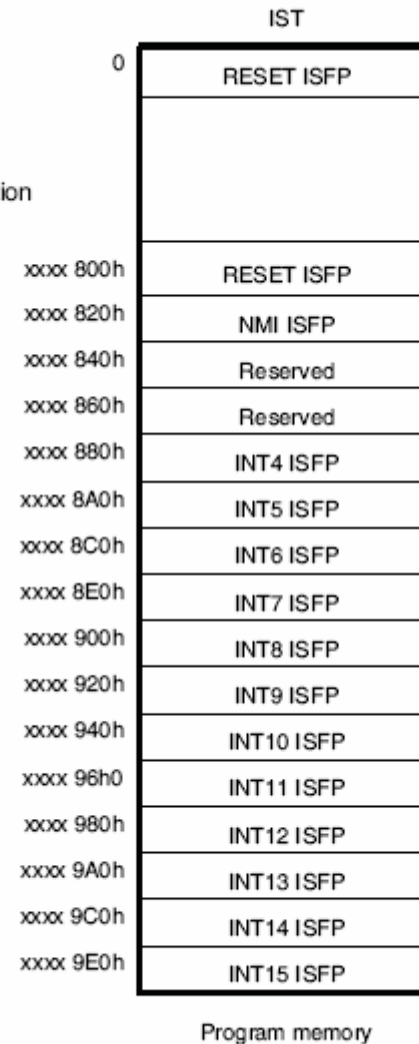
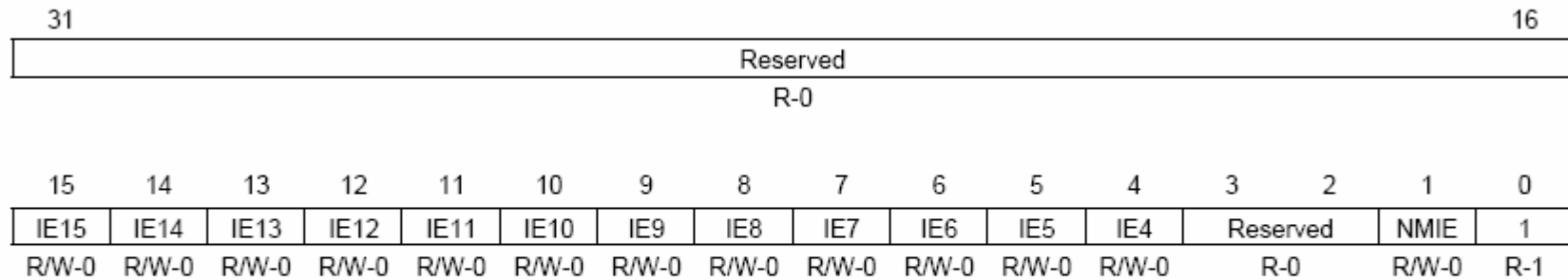


Figure 2-8. Interrupt Enable Register (IER)

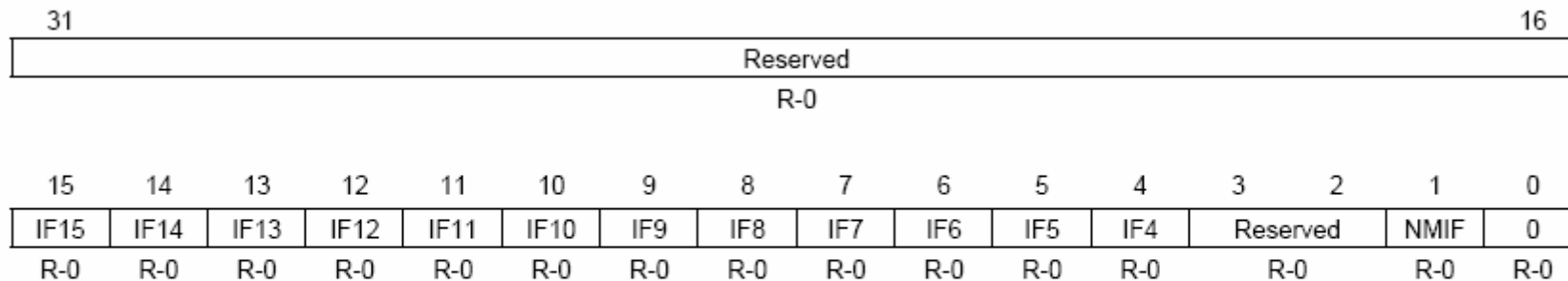


LEGEND: R = Readable by the MVC instruction; W = Writeable by the MVC instruction; -n = value after reset

Table 2-12. Interrupt Enable Register (IER) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15-4	IE _n	0	Interrupt enable. An interrupt triggers interrupt processing only if the corresponding bit is set to 1. Interrupt is disabled.
		1	Interrupt is enabled.
3-2	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
1	NMIE	0	Nonmaskable interrupt enable. An interrupt triggers interrupt processing only if the bit is set to 1. The NMIE bit is cleared at reset. After reset, you must set the NMIE bit to enable the NMI and to allow INT15-INT4 to be enabled by the GIE bit in CSR and the corresponding IER bit. You cannot manually clear the NMIE bit; a write of 0 has no effect. The NMIE bit is also cleared by the occurrence of an NMI. All nonreset interrupts are disabled.
		1	All nonreset interrupts are enabled. The NMIE bit is set only by completing a B NRP instruction or by a write of 1 to the NMIE bit.
0	1	1	Reset interrupt enable. You cannot disable the reset interrupt.

Figure 2-9. Interrupt Flag Register (IFR)



LEGEND: R = Readable by the MVC instruction; -n = value after reset

Table 2-13. Interrupt Flag Register (IFR) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15-4	IF n	0 1	Interrupt flag. Indicates the status of the corresponding maskable interrupt. An interrupt flag may be manually set by setting the corresponding bit (IS n) in the interrupt set register (ISR) or manually cleared by setting the corresponding bit (IC n) in the interrupt clear register (ICR). Interrupt has not occurred. Interrupt has occurred.
3-2	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
1	NMIF	0 1	Nonmaskable interrupt flag. Interrupt has not occurred. Interrupt has occurred.
0	0	0	Reset interrupt flag.

□ Recapitulare – registrele folosite pentru întreruperi

○ CSR – Registrul de control al stării (Control Status Register)

- permite setarea sau invalidarea globală a întreruperilor

○ ICR – Registrul de ștergere a întreruperilor (Interrupt Clear Register)

- permite ștergerea manuală a indicatorilor în registrul IFR

○ IER – Registrul de validare a întreruperilor (Interrupt Enable Register)

- permite validarea întreruperilor

○ IFR – Registrul indicator a întreruperilor (Interrupt Flag Register)

- indică starea întreruperilor

○ IRP – Registrul pointer la revenire din întrerupere (Interrupt Return Pointer Register)

- conține adresa de revenire utilizată la revenirea dintr-o întrerupere mascabilă. Revenirea se realizează folosind instrucțiunea **B IRP**.

- ISR – Registrul de setare a întreruperii (Interrupt Set Register)
 - permite setarea manuală a indicatorilor din registrul IFR
- ISTP – Registrul pointer la tabelul de tratare a întreruperii (Interrupt Service Table Pointer Register)
 - pointer la începutul tabelului de tratare a întreruperii
- NRP – Registrul pointer la revenirea dintr-o întrerupere nemascabilăde (Nonmaskable Interrupt Return Pointer Register)
 - conține adresa de revenire utilizată la revenirea dintr-o întrerupere nemascabilă. Revenirea se realizează folosind instrucțiunea **B NRP**.

□ Validarea și invalidarea globală a întreruperilor

- Registrul de control al stării CSR conține doi biți care controlează întreruperile: **GIE** și **PGIE**.
- Bitul de validare globală a întreruperilor GIE realizează validarea sau învalidarea globală a întreruperilor mascabile. Este bitul CSR.0.
- Dacă GIE = 1L, atunci sunt validate global întreruperile mascabile; dacă GIE = 0L, atunci sunt invalidate global întreruperile mascabile (se realizează cu instrucțiunea **MVC**).
- CPU detectează cerearea de întrerupere în paralel cu execuția instrucțiunii. După ce CPU începe tratarea întreruperii valoarea bitului GIE se copiază în bitului PGIE (CSR.1). Apoi, CPU setează GIE = 0L pentru a preveni apariția unei alte întreruperi mascabile înainte de salvare a stării procesorului. La revenirea din subrutina de tratare a întreruperii (cu instrucțiunea **B IRP**) CPU copiază valoarea bitului PGIE în bitul GIE (va avea valoarea dinaintea intrării în subrutina de tratare a întreruperilor).

○ Exemplu invalidare globală a întreruperilor mascabile

```
MVC   CSR,B0           ; get CSR
AND   -2,B0,B0        ; get ready to clear GIE
MVC   B0,CSR          ; clear GIE
```

○ Exemplu validare globală a întreruperilor mascabile

```
MVC   CSR,B0           ; get CSR
OR    1,B0,B0         ; get ready to set GIE
MVC   B0,CSR          ; set GIE
```

□ Validarea și invalidarea unei întreruperi individuale

- Validarea și invalidarea întreruperii individuale se realizează folosind registrul validare a întreruperii (IER).
- Bitul NMIE (IER.1) = 0L invalidează toate întreruperile cu excepția lui Reset.
- La inițializarea procesorului NMIE = 0L.
- După inițializarea procesorului bitul NMIE trebuie setat pe 1L pentru a valida întreruperea nemascabilă și pentru a permite validarea întrerupeilor INT4-INT15 cu ajutorul bitului GIE și a bitului corespunzător din IER.
- Bitul NMIE nu poate fi setat pe 0L manual. Este setat pe 0L prin apariția întreruperii mascabile.
- Dacă NMIE are valoarea 0L el poate trece pe valoarea 1L prin programare sau în urma execuției instrucțiunii B NRP.

○ Exemplu validare întrerupere individuală INT9

```
MVK    200h,B1    ; set bit 9
MVC    IER,B0     ; get IER
OR     B1,B0,B0   ; get ready to set IE9
MVC    B0,IER     ; set bit 9 in IER
```

○ Exemplu invalidare întrerupere individuală INT9

```
MVK    FDFFh,B1   ; clear bit 9
MVC    IER,B0
AND    B1,B0,B0   ; get ready to clear IE9
MVC    B0,IER     ; clear bit 9 in IER
```

□ Starea întreruperilor

- Starea întreruperii în așteptare este memorată în registrul indicator întrerupere (IFR).
- Registrul IFR conține starea întreruperilor INT4-INT15 și a întreruperii NMI.
- Dacă o întrerupere apare, atunci bitul corespunzător ei din registrul IFR este 1L. În caz contrar bitul este 0L.
- Pentru citirea registrului IFR se utilizează instrucțiunea MVC.

□ Setarea și ștergerea întreruperilor

- Registrul de setare întreruperi (ISR) și registul de ștergere întreruperi (ICR) permit setarea și ștergerea manuală a întreruperilor mascabile din registrul IFR.
- Scrierea 1L în unul din biții IS4-IS15 din registrul ISR conduce la setarea indicatorului de întrerupere corespunzător din IFR. Similar scrierea 1L în unul din biții IC4-IC15 din registrul ICR conduce la ștergerea indicatorului de întrerupere corespunzător din IFR.
- Scrierea 0L în biții registrelor ISR și ICR nu are nici un efect.

○ Exemplu setare întrerupere individuală INT6 și citirea registrului IFR

```
MVK    40h, B3  
MVC    B3, ISR  
NOP  
MVC    IFR, B4
```

○ Exemplu ștergere întrerupere individuală INT6 și citirea registrului IFR

```
MVK    40h, B3  
MVC    B3, ICR  
NOP  
MVC    IFR, B4
```

□ Revenirea din subrutina de tratare a întreruperii

- Cu ajutorul pointerilor pentru revenirea dintr-o întrerupere este restaurată starea inițială.
- Registrul pointer de revenire din întreruperea nemascabilă NRP conține pointerul de revenire care indică CPU locația pentru continuarea execuției programului după tratarea întreruperii NMI.
- Revenirea din întreruperea nemascabilă (NMI)

```
B    NRP    ; return, sets NMIE  
NOP  5     ; delay slots
```

○ Registrul pointer de revenire din întrerupere IRP conține pointerul de revenire care indică CPU locația pentru continuarea execuției programului după tratarea întreruperii mascabile.

○ Revenirea dintr-o întrerupere mascabilă (INT)

```
B      IRP      ; return, moves PGIE to GIE
NOP    5        ; delay slots
```