Architecture version 2.0

Architecture Specification

ARM®

, , . .

. ( )

ARM Generic Interrupt Controller

Release Information

T e f c a e a e bee ade d c e .

Change History

Date	Issue	Confidentiality	Change
23 Se e be 2008	A	N -C fde a	F e ea e f e 1.0
13 J e 2011	B	N -C fde a	F e ea e f e 2.0
26 J 2013	B.b	N -C fde a	Re- e ea e f e B e P e a N ce

Status of Issue B.b of this document

I e B.b f d c e a e- e f e B c a e da ed P a a N ce f ed c e .
Be d a e f f ed c e e c a e be ee e B a d e B.b a e:

- C a e e a e f e e e e be , c da e , a d ID c de.
- C a ed a e be , beca e f e e P a a N ce.
- A a e e A e d C*R* a e e a e ec ca c a e be ee e B a d e B.b.

Proprietary Notice

ARM GENERIC INTERRUPT CONTROLLER (GIC) ARCHITECTURE SPECIFICATION LICENCE

THIS END USER LICENCE AGREEMENT ("LICENCE") IS A LEGAL AGREEMENT BETWEEN YOU (EITHER A SINGLE INDIVIDUAL, OR SINGLE LEGAL ENTITY) AND ARM LIMITED ("ARM") FOR THE USE OF THE RELEVANT GIC ARCHITECTURE SPECIFICATION ACCOMPANYING THIS LICENCE. ARM IS ONLY WILLING TO LICENSE THE RELEVANT GIC ARCHITECTURE SPECIFICATION TO YOU ON CONDITION THAT YOU ACCEPT ALL OF THE TERMS IN THIS LICENCE. BY CLICKING "I AGREE" OR OTHERWISE USING OR COPYING THE RELEVANT GIC ARCHITECTURE SPECIFICATION YOU INDICATE THAT YOU AGREE TO BE BOUND BY ALL THE TERMS OF THIS LICENCE. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENCE, ARM IS UNWILLING TO LICENSE THE RELEVANT GIC ARCHITECTURE SPECIFICATION TO YOU AND YOU MAY NOT USE OR COPY THE RELEVANT GIC ARCHITECTURE SPECIFICATION AND YOU SHOULD PROMPTLY RETURN THE RELEVANT GIC ARCHITECTURE SPECIFICATION TO ARM.

"LICENSEE" ea Y a d S b da e .

"S b da " ea , f Y a e a e e , a c a e a f e a e e eaf e ed c ed, d ec d ec , b Y . A c a a be a S b da f e e d d c c c e .

1. S b ec e f C a e 2, 3 a d 4, ARM e eb a LICENSEE a e e a , -e c e, - a fe ab e, a f ee, d de ce ce :
   a. e a d c e e e a GIC A c ec e S ec f ca f e e f de e a d a de e ed d c a c e e e a GIC A c ec e S ec f ca ;
   b. a fac e a d a e a fac ed d c c e e : ( ) a e bee c ea ed b f LICENSEE de e ce ce a ed C a e 1a; ( ) c a e a d c ( ) c a bee c ea ed b a d a ( ) de a ce ce a ed b ARM C a e 1a f c d a ARM GIC A c ec e S ec f ca L ce ce; a d
   c. ffe e , e , e ed b e d c c a e e bee ( ) c ea ed b f LICENSEE de e ce ce a ed C a e 1a; ( ) a fac ed b f LICENSEE de e ce ce a ed C a e 1b.
2. LICENSEE e eb a ee a e ce ce a ed C a e 1 b ec e f e c :
   a. e e a d c c ea ed de C a e 1a a fac ed de C a e 1b c a a ea e ce c e c a e e bee ( ) de e ed b f ARM; ( ) de e ed de ce ce f ARM;
   b. e ce ce a ed C a e 1c a e e d a f c fa d c a e f c a a f e e e a GIC A c ec e S ec f ca ; a d
   c. a ed LICENSEE b ce e e a ed LICENSEE de A ee e .

3. E ce a ec f ca ce ed acc da ce C a e 1, LICENSEE ac e , e e e a ARM ec a e ec a e e b d ed e e . I e e a e ce ce a ed acc da ce C a e 1 be c ed a a LICENSEE, e e b ca , e e e e, a ce ce e a ARM ec e ce e e e a GIC A c ec e S ec f ca .

4. THE RELEVANT GIC ARCHITECTURE SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF SATISFACTORY QUALITY, MERCHANTABILITY, NONINFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE.

5. N ce ce, e e , ed e e, a ed LICENSEE, de e f C a e 1, e e ARM ade a e c ec e e e a GIC A c ec e S ec f ca a d c ba ed e e . N C a e 1 a be c ed a a f LICENSEE a e a e e e a be a f f ARM e ec f e e e a GIC A c ec e S ec f ca .

6. T L ce ce a e a f ce e a ed b b ARM. W e d ce a f e f LICENSEE b eac f a f e e a d c d f L ce ce e ARM a e a e L ce ce ed a e e ce Y . Y a e a e L ce ce a a e. U e e a f L ce ce b Y b ARM LICENSEE a e e e a GIC A c ec e S ec f ca a d de

ARM Generic Interrupt Controller Architecture Specification

Preface

T eface d ce e *ARM G I C A S* . I c a e f ec :

A a e

a e

C a e

A a e

F a e .

About this specification

T ec f ca de c be e *ARM G I C* (GIC) a c ec e.

T d c e , efe e ce *GIC GIC* efe a de ce a e e GIC a c ec e. U e e c e a e c ea a a efe e ce a IMPLEMENTATION DEFINED fea e f e de ce, e e efe e ce de c be e e e e f ec f ca .

Intended audience

T e ec f ca e f e a a de , e e , a e GIC a a e f ARM-c a e e a f e ce e e a c e ce e .

T e ec f ca a e a e a e e e e ce f ARM d c . I d e a e e e ce f e GIC.

Using this specification

T ec f ca a ed e f c a e :

Chapter 1 *I*

Read f a e e f e GIC, a d f a ab e e ed d c e .

Chapter 2 *GIC P*

Read f a de c f e a e face a d c e f e GIC. T e c a e a d ce e e a e, a e e e a .

Chapter 3 *I H P*

Read f a de c f e e e e f e a d , a d e e c e e f a GIC.

Chapter 4 *P M*

Read f a de c f e D b a d CPU e face e e .

Chapter 5 *GIC S*

Read f a de c f e GIC V a a E e e e a f a GIC a ce e a ce a a .T c a e c de a de c f e a e de f e a e face c a d a CPU e face e e .

Appendix A *P I*

Read f a de e e d c de f c def ed ec f ca .

Appendix B *R N*

Read f a de c f e d ffe e ce e e e a e ea e de c f e GIC a c ec e, a d f a a abe c de f e e e a e .

Appendix C *R*

Read f a de c f e ec ca c a e be ee e ea ed e f b .

G Read f def f e e ed b .

Conventions

T e f ec de c be c e a b ca e:

G
S
N
P .

General typographic conventions

T e a ca c e a e:

italic I d ce ec a e , de e e a c - efe e ce a d c a ,
a a e.

bold De e a a e , a d ed f e de c e , e e a a e.

`monospace` U ed f a e b e a de c , e d c de, a d ce c de e a e .
A ed e a e f c e c a d f efe e ce e e
a ea a e b e a de c , e d c de, a d ce c de e a e .

SMALL CAPITALS U ed f a fe e a a e ec f c ec ca ea , a d a e c ded e
G a .

Colored text I d ca e a . T ca be:
- a URL, f e a e, `http://infocenter.arm.com`
- a c - efe e ce, a c de e a e be f e efe e ced f a f
e c e a e, f e a e, *D C R , GICD CTLR* a e 4-85
- a , a c a e a e d , a a e , e ec f e
d c e a def e e c ed e , f e a e *B* GICD_CTLR.

Signals

I e e a ec f ca d e def e ce a , b d e c de e a e a e a d
ec e da . T e a c e a e:

Signal level T e e e f a a e ed a de e d e e e a ac e-HIGH
ac e-LOW. A e ed ea :
HIGH f ac e-HIGH a

Additional reading

T ec e e a b ca f ARM a d d a e .

See e I f ce e , http://infocenter.arm.com, f acce ARM d c e a .

ARM publications

ARM A R M , ARM 7-A ARM 7-R (ARM DDI 0406), e C a e .

Other publications

T e f b a e efe ed a a , de e f a :
JEDEC S d S a e Tec A c a , *S M I C* , JEP106.

Feedback

ARM e c e feedbac d c e a .

Feedback on this specification

If a e c e e c e f ec f ca , e d e- a errata@arm.com. G e:
e e
e be , ARM IHI 0048B.b
e a e be c c e a
a c c e e a a f c e .

ARM a e c e e e a e f add a d e e .

Introduction

T c a e e a e e f e GIC a d f a ab e e ed d c e . I c a e f ec :

A G I C a e 1-14

S E a e 1-16

a e 1-17

T a e 1-18.

1.1 About the Generic Interrupt Controller architecture

T e *G I C* (GIC) a c ec e def e :
e a c ec a e e e f a d a e ce f a ce c ec ed a GIC
a c e c e a e face a cab e ce ce e .

Note

T e a c ec e de c be a GIC de ed f e e e ce a c e ARM A a d R a c ec e f e . H e e e GIC a c ec e d e ace a e c e ce ed a e e a f e GIC.

T e GIC a ce a ed e ce f a d a a e a e a c de a ea e ce . I de :
e e f a a e ce , e be a , a d e e e ce f :

- G e ARM a c ec e Sec E e
- G e ARM a c ec e V a a E e
- G e ab , d ab , a d e e a ce e f a d a e (e e a) e ce
- G S f a e- e e a ed I e (SGI)
- G e a a d a
- G ce a d ce e e
- G a e e e e- a a e e e e .

T e GIC c de f c a a :
c f eac e a e e G 0 G 1
a G 0 e e a e ce e e e IRQ e FIQ e ce e e
a G 1 e e a e ce e IRQ e ce e e
a f ed c e e f a d e f G 0 a d G 1 e
a c d f e c f a f e G 0 e .

Note

I e e e a GIC 2 e e a a d GIC 1 e e a a c de e GIC Sec E e , ee *C 2.0 S* a e 1-15.

I a e e a e IRQ a d FIQ e e e c e d e IRQ a d FIQ a c e ce a a e ed b a a a f e ARM a c ec e e ce e *M* (M- f e). F e f a ab IRQ, FIQ, a d a c e ce , ee e *ARM A R M , ARM 7-A ARM 7-R* .

1.1.1 GIC architecture specification version

T ec f ca def e e 2.0 f e GIC a c ec e (GIC 2), a d a de c be e 1.0 f e a c ec e (GIC 1).

T e GIC a c ec e ec f ca e de e de f e e , a a d e de c , ed f ARM d c e ea e .

1.1.2 Changes in version 2.0 of the Specification

Ve 2.0 f e A c ec e S ec f ca c a e f c a e a d add e 1.0:

1. T e add f e a GIC V a a E e , a e e e a f e GIC a e a ce a a . F e f a , ee a e 1-17.
2. A c a e e a c ec a a f . I e , a d e ab e FIQ a G 0 e , a e ded:
   a GIC 2 e e a
   a a f e a Sec E e GIC 1 e e a .

   Note

   I e 1.0 f e S ec f ca , e e e ed a e c a f ca f e a Sec e N - ec e, ee e 7 f .
3. T e add f a e e e a a e e e e . F e f a , ee *P , GIC 2* a e 2-31.
4. T e add f f e a e a d e e f a GIC a e, f e - d , c e c , c d a ac e c e c a e a a a . T ea a e a e a ead- GIC 1 bec e ead/ e GIC 2. F e f a , ee *P GIC* a e 4-155.
5. T e add f a e c e a e , *P* a d . F e f a , ee *P* a e 3-38.
6. T e add f c d ab e e f a d f e ac e a a c ec ed ce e f a d f e f e GIC a ce a d ab ed. F e f a ee *I , GIC 2* a e 2-27.
7. C a e e e ed de c be e e fea e f e GIC 1 Sec E e , ca f a e e fea e ca be ed e e f c a a e a ed e c e f e ARM Sec E e e e a ARM ce .

   Note

   A d ca ed e 2, e e fea e f e GIC 1 Sec E e a e c ded a GIC 2 e e a . T a , GIC 2 e a e a f e a Sec E e .

T e e c a e c de e a e I e Sec Re e I e G Re e . T e e e e a a e e , G 0 a d G 1. I ec f c c e , ca e a GIC a e e e GIC Sec E e c ec ed a ARM ce a e e e ce Sec E e , G 0 e a e Sec e e a d G 1 e a e N - ec e e . F e f a , ee *S E* a e 1-16.

1.2 Security Extensions support

T e ARM ce Sec E e a e a e e e ARM 7-A a c ec e f e. T ea IMPLEMENTATION DEFINED e e a ARM 7-A e e a c de e Sec E e . T e ARM Sec E e fac a e e de e e f ec e a ca b :

e a a d a e ec fea e e a c ec e

d Sec e a e ace a acce ed b e acce e e Sec e a e

d N - ec e a e ace a acce ed b e acce e e N - ec e a e.

See *P S N - GIC* a e 1-20 f e f a .

W e a GIC a e e e GIC Sec E e c ec ed a ce a e e e ARM Sec E e :

G 0 e a e Sec e e , a d G 1 e a e N - ec e e .

T e be a f ce acce e e e e GIC de e d e e e acce Sec e N - ec e, ee *P S N - GIC* a e 1-20.

E ce e e d c e e c d ca e e e, e acce GIC e e :

G a N - ec e ead f a e e f e d d a e f a f a Sec e e e e

G e GIC e a N - ec e e a e e f e d d a e f a f a Sec e e .

N - ec e acce e ca ead e f a c e d N - ec e e . Sec e acce e ca ead e f a c e d b N - ec e a d Sec e e .

Sec e e f a e d d a def e eac e e e d e a e e Sec e N - ec e.

A N - ec e e a a IRQ e e e a a e ce .

A Sec e e ca a e e a IRQ a FIQ e e e a a e ce .

Sec e f a e ca a a e e ce ec e e b f e fe e ce f N - ec e f a e. See *C S N -* a e 3-69 f e f a .

Sec e e a e bac a d -c a b e f a e e f e e Sec E e . See *S IRQ FIQ S E* a e 3-70 f e f a .

1.3 Virtualization support

T e ARM ce V a a E e a e a e e ARM 7-A a c ec e f e. T ea IMPLEMENTATION DEFINED e e a ARM 7-A e e a c de e V a a E e .

T e ce V a a E e de a d a e f a e N - ec e a e fa VMSA 7 e e a . T e e e e e fa a ac e , a e e , c e e a e .

W e e e ed a ce a ce e , e ce V a a E e e a ac e a e ce .

I e a d a a c de a a a a e e a . T e e ca e e a d e a ca e e f, e e a e a c e d a e a a ed a a ac e. I a b e f e e e e a e a e a d c e d ca e .

GIC 2 e e d e GIC a c ec e c de e GIC V a a E e . T e e e e e a d f a e , add ca e , a e a ce a a . A e a e f c a e e e e a GIC e a ed ce a e e e ARM ce V a a E e . T e GIC V a a E e de ec a e e e e ead f e a ac e . See C a e 5 *GIC S* f e f a .

Note

A ce a e e e ARM V a a E e a e e e ARM Sec E e .

A GIC a e e e GIC V a a E e e ed e e e GIC Sec E e .

1.4 Terminology

T e f ec def e a c ec a e ed ec f ca :

I

I

M a e 1-19

S a e 1-20

P *S* *N* - *GIC* a e 1-20

B a e 1-20.

See a *GIC* a e 4-74.

1.4.1 Interrupt states

T e f a e a a eac e face be ee e GIC a d a c ec ed ce :

Inactive A e a ac e e d .

Pending A e f a ce e GIC a ec ed a a e ed a d a e, e e a ed b f a e, a d a be e ced b a a e ce .

Active A e f a ce e GIC a a bee ac ed ed b a ce , a d be e ced b a c e ed.

Active and pending A ce e c e e a d e GIC a a e d e f e a e ce.

1.4.2 Interrupt types

A de ce a e e GIC a c ec e ca a a e e f e f e :

Peripheral interrupt T a e a e ed b a a e GIC. T e GIC a c ec e def e e f e f e e a e :

Private Peripheral Interrupt (PPI)
T a e e a e a ec f c a e ce .

Shared Peripheral Interrupt (SPI)
T a e e a e a e D b ca e a f a ec f ed c b a f ce .

Eac e e a e e e :

Edge-triggered
T a e a a e ed de ec f a ed e f a e a a d e, e a d e f e a e f e a, e a a e ed c ea ed b e c d def ed b ec f ca .

Level-sensitive
T a e a a e ed e e e e a e e ac e, a d dea e ed e e e e e e ac e.

Note

W e a e e - e e e a e ed a e e GIC e d , ac e a d e d . If e e e a dea e e e a f a ea e GIC e e e e d a e f e e . F e f a ee *I* a e 3-41.

Software-generated interrupt (SGI)

T a e e e a ed b f a e a GICD_SGIR e e e GIC. T e e e SGI f e ce c ca .

A SGI a ed e- e ed e e . T e f a e e f e e e a e e ed e a f e e e e a .

W e a SGI cc a ce e e a , e CPUID f e d e I e Ac ed e Re e , GICC_IAR, e A a ed I e Ac ed e Re e , GICC_AIAR, de f e e ce a e e ed e e .

I a e e a a c de e GIC V a a E e :

e a SGI cc , a a e e e e e GIC a a E e e ab e e e e ce be e ed e G e OS, a e ed b e GIC ec f ca

b e a a e e e e GIC V a a E e , a e ca e e a e a a e a a ac e a a SGI.

See *S* - a e 5-165 a d *L R* , *GICH LR* a e 5-176 f e f a .

Virtual interrupt I a GIC a e e e GIC V a a E e , a e a a e a a ac e a ce , a d ca a ed e ce b e c ec ed a CPU e face. F e f a , ee *A GIC* a e 2-22.

Maintenance interrupt

I a GIC a e e e GIC V a a E e , a e e - e e e a ed a e e e e , c a a a c a f e bec e ab ed d ab ed. See *M* a e 5-164 f e f a .

1.4.3 Models for handling interrupts

Note

W e de c b e GIC e a d de , e e 1-N a d N-N d c e d e a e a ca e f e e 1:N a d N:N.

I a ce e e a , e e a e de f a d e :

1-N model O e ce a de e . T e e e a ec a de e e c ce a de a e a a ed a e e a e ce .

Note

T e ARM GIC a c ec e d e a a ee a a 1-N e e e ed :

- G a ce ed e a e ce
- G a e ab ed e face, e e a ea e e face e ab ed.

A 1-N e be e e ed a e face e e e ce a a ed e e e e , ee *I 1-N* a e 3-41.

N-N model A ce ece e e e de e de . W e a ce ac ed e e e , e e e d a e c ea ed f a ce . T e e a e d f e e ce .

See *H* a e 3-35 f e f a .

1.4.4 Spurious interrupts

I b e a a e a e GIC a a ed a ce e e ed. If a e , e e ce ac ed e e e , e GIC e a ec a I e ID a de f e e e a a . E a e ea f e a e:

e ce ac ed a e :

- G f a ec a e e f e e
- G f a e d ab e e e
- G f a ec a e e ce a e e a e

f a 1-N e , a e a e ce a e ac ed ed a e .

1.4.5 Processor security state and Secure and Non-secure GIC accesses

A ce a e e e ARM Sec E e a a ec a e, e e Sec e N - ec e:

a ce N - ec e a e ca a e N - ec e acce e a GIC

a ce Sec e a e ca a e b Sec e a d N - ec e acce e a GIC

f a e N - ec e a e de c bed a N - ec e f a e

f a e Sec e a e de c bed a Sec e f a e.

F e f a ab e e e a f e Sec E e a ce ee e *ARM A R M* , *ARM 7-A* *ARM 7-R* .

A ce e a GIC a e e e Sec E e c de e e ce a d e e e Sec E e . S c a ce e e ed a e e:

a e Sec e acce e e GIC, ea a f a e e ce Sec e f a e a ca a e Sec e acce e e GIC

a e N - ec e acce e e GIC, ea a f a e e ce N - ec e f a e.

1.4.6 Banking

B a a ec a ea ARM a c ec a ec f ca :

Interrupt banking

I a ce e e a , f PPI a d SGI , e GIC ca a e e e e a e e ID. S c a e ca ed a , a d de f ed e b ec b a f e ID a d a c a ed CPU e face. F e f a ee *I ID* a e 2-24.

Register banking

Re e ba efe e e ec e f a e e a e a e add e . T cc :

a ce e e a , de e a a ec e f eac ce f e e c e d ba ed e

a GIC a e e e Sec E e , de e a a e Sec e a d N - ec e c e f e e e e .

F e f a ee *R* a e 4-77.

GIC Partitioning

T c a e de c be e a c ec a a f e a GIC e face a d c e , a d d ce e f c a f e a GIC c e , e *D* a d e *CP* . I c a e f ec :

2.1 About GIC partitioning

T e GIC a c ec e ca a D b b c a d e e CPU e face b c . T e GIC V a a E e add e e a CPU e face e GIC. T e ef e, a F e 2-1 a e 2-23 , e ca a f e GIC a f :

Distributor T e D b b c e f e a a d d b e CPU e face b c a c ec e ce e e .

T e D b b c e e a e de f ed b e GICD_ ef .

CPU interfaces Eac CPU e face b c e f a a d ee a d f a c ec ed ce e e .

CPU e face b c e e a e de f ed b e GICC_ ef .

W e de c b a GIC a c de e GIC V a a E e , a CPU e face e e ca ed a *CP* , a d b e c f a a CPU e face.

Virtual CPU interfaces

T e GIC V a a E e add a a CPU e face f eac ce e e . Eac a CPU e face a ed e f b c :

Virtual interface control

T e a c e f e a e face c b c e GIC a e face c e e , a c de a f ac e a d e d a e f ec e a ac e e c ec ed ce . T ca , e e e a e a a ed b e e a a ce .

V a e face c b c e e a e de f ed b e GICH_ ef .

Virtual CPU interface

Eac a CPU e face b c de ca a f a e e c ec ed ce . T e ARM ce V a a E e a e e e e c e a ac e a ce . T e GIC a CPU e face e e , acce ed b e a ac e, de e c a d a f a f e a e . T e f a f e e e e a e f a f e ca CPU e face e e .

V a CPU e face b c e e a e de f ed b e GICV_ ef .

Note

T e a CPU e face d e e e a a e e f c a de c bed *P , GIC 2* a e 2-31.

A GIC ca e e e CPU e face , be ed f 0-7. I a GIC a e e e GIC V a a E e , a CPU e face be c e d e CPU e face be , a CPU e face 0 a d a CPU e face 0 c ec e a e ce .

T de e e a f e GIC ce ce e e , a d e GIC V a a E e e e d a ce a a a , c , N - ec e a e:

A G e OS a a ac e

A e e b e f c be ee a ac e . T c c de c e a e e d e GIC a e face c e e .

Eac b c de a f e GIC a e de , a d:

e a e de e e a e a e f eac e e ed CPU e face.

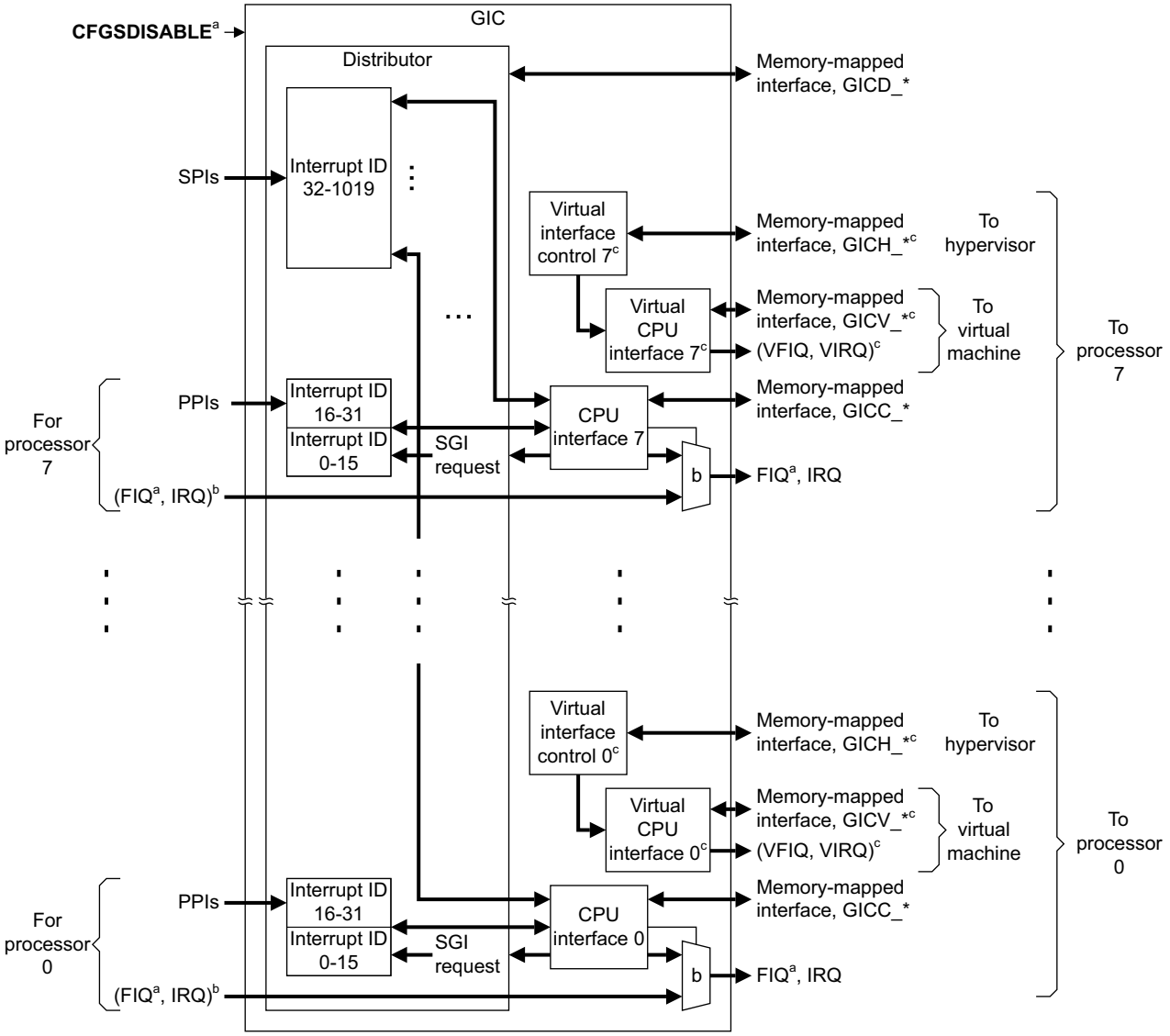
e a e de f a a CPU e face e e a e a e a e de f a ca CPU e face.

Note

T e a f e GIC de c bed ec a a c ec a ab ac . W e e e eb c a e e e ed e a a e c b ed IMPLEMENTATION SPECIFIC.

I a GIC a e e e GIC Sec E e a ce e , a CPU e face ca be e e ed a ece e :

G b Sec e a d N - ec e acce e
G Sec e acce e
G N - ec e acce e .



[a] In GICv1, applies only if Security Extensions are implemented
[b] Optional input and bypass multiplexer, see text
[c] Applies only to GICv2 with Virtualization Extensions

Figure 2-1 GIC logical partitioning

T e e a de f c a e , a d C a e 3 *I H P* a d C a e 4 *P M* , de c be e GIC e GIC V a a E e . C a e 5 *GIC S* de c be e fea e added b e GIC V a a E e .

2.2 The Distributor

T e D b ce a e a e ce , de e e e f eac e , a d f eac CPU e face f a d e e e e e face, f a a d ee a d .

T e D b de a a e face f :

- G ba e ab e f a d f e e CPU e face .
- E ab d ab eac e .
- Se e e e f eac e .
- Se e a e ce f eac e .
- Se eac e e a e be e e - e e ed e- e ed.
- Se eac e a e e G 0 G 1.

Note

F GIC 1, e e a G 0 G 1 b e e e e a c de e GIC Sec E e .

F a d a SGI e e a e ce .

I add , e D b de :

b f e a e f eac e a ec a f f a e e c ea e e d a e f a e e a e .

2.2.1 Interrupt IDs

I e f ce a e de f ed *ID* . Eac CPU e face ca ee 1020 e . T e ba f SPI a d PPI c ea e e a be f e ed b e D b .

T e GIC a e ID be ID0-ID1019 a f :

I e be ID32-ID1019 a e ed f SPI .

I e be ID0-ID31 a e ed f e a a e a e a CPU e face. T e e e a e ba ed e D b .

A ba ed e e e e e D b ca a e e e e a e ID. A ba ed e de f ed e b ID be a d a c a ed CPU e face be . Of e ba ed e ID :

- G ID0-ID15 a e ed f SGI
- G ID16-ID31 a e ed f PPI

I a ce e :

- G A PPI f a ded a a c a CPU e face, a d a e a e face. I e f a CPU e face e D b d e c de PPI a e a e e e face .
- G Eac c ec ed ce e a SGI b e GICD_SGIR e D b . Eac e ca e e a e SGI e a e ID a a e e ce .

  I e D b , a SGI de f ed e b e c b a f e be , ID0-ID15, e a e ce ID, CPUID0-CPUID7, a d e *ID*, CPUID0-CPUID7, f e ce a ed e SGI. W e e CPU e face c ca e e e ID a a e ed ce , a de e ce ce ID, a e a e ed ce ca e de f e SGI.

  SGI ba ea e GIC ca a d e e SGI a e , e ce c f c .

  T e D b e a e e e GICD_SGIR a f a ce a c ec ed e f e CPU e face . H e D b de e e e ce ce ID f a ce e GICD_SGIR IMPLEMENTATION SPECIFIC.

I a ce e , e e d c be ee a ed a d a e e , beca e a e a e b e e ce . I ca e e ce ce ID a e 0.

I e be ID1020-ID1023 a e e e ed f ec a e , ee *S* a e 3-43.

S e f a e e e f eac e . T de e de f e e ID be .

I a e a e e e ARM Sec E e , a c e de f e a e a be ee ce , ARM ec e d a a ce e e e:

ID0-ID7 f N - ec e e

ID8-ID15 f Sec e e .

2.3 CPU interfaces

Eac CPU e face b c de e e face f a ce a c ec ed e GIC. Eac CPU e face de a a e face f :

e ab e a f e e e ce
ac ed a e
d ca c e f e ce f a e
e a e a f e ce
def e ee c f e ce
de e e e d e f e ce .

W e e ab ed, a CPU e face a e e e d e f c ec ed ce a d de e e e e e a f a e e e e ce . T de e e e e a e e e e ce , e CPU e face c de e e a a d e ee e f e ce . A a e, e c ec ed ce ca ead e f e ac e e f GICC_HPPIR, a CPU e face e e .

T e ec a f a a e e ce IMPLEMENTATION DEFINED.

Note

O ARM ce e e a , e ad a ec a f a a e e e b a e **nIRQ** **nFIQ**.

T e ce ac ed e e e e e b ead e CPU e face I e Ac ed e Re e . T ead e e f:

T e ID be f e e e d e , f a e f ff c e f be a ed e ce . T e a e e a e ac ed e.

E ce a , a ID be a d ca e a .

W e e ce ac ed e e e a e CPU e face, e D b c a e e a f e e f e d e e ac e, ac e a d e d . A e CPU e face ca a a e e e ce , ee e a a e ac e e ce . If e e e d e ff c e f a e ce , e e face dea e e e e e a e ce .

W e e e a d e e ce a c e ed e ce f a e , e e CPU e face d ca e e c e . T e e a e a e e c e :

d , ea e f e ce ed e ca e e e e a f a e e e ce

e deac a , ea e D b e e e ac e a e f e e .

I a GIC 1 e e a , e e a e a a a e e e, e e ce e e CPU e face E d f I e e e.

I a GIC 2 e e a , e GICC_CTLR.EOI de b de e e e e :

e a e a e e e, e e ce e e CPU e face E d f I e e e

e a e a e e a a ed, a :

G d a e e e ce e e CPU e face E d f I e e e

G e deac a a e a e, e e ce e e CPU e face Deac a e I e e e.

F e f a , ee *P* a e 3-38.

2.3.1 Interrupt signal bypass, and GICv2 bypass disable

I a GIC e e a , a CPU e face a c de e a b a , a , e e a f a e b e e face d ab ed, a e e ac e a a ed e e e e ce , b a e GIC f c a .

F e 2-2 e e e a f e a b a a GIC 1 e e a a d e c de e GIC Sec E e .

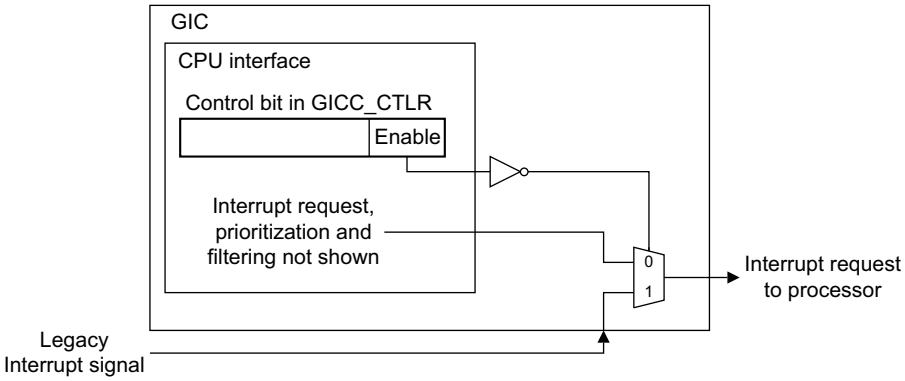


Figure 2-2 Interrupt signal bypass, GICv1 without Security Extensions

F e 2-2 e e e e a f e a b a . I e GIC e e a , e a b a e c ca ed:

A GIC 1 e e a a c de e GIC Sec E e e , a d e e f FIQ e a G 0 e . *I , GIC 1 GIC S E* de c be e e e a f e b a c a e e a .

If a GIC 2 e e a e a b a , e e a e de a a GIC 1 e e a a c de e GIC Sec E e , b a ded ab e b f e e a b a e a . F e f a ee *GIC 2* , a e 2-28.

Note

Ma ARM ce , c d ce a e e e ARM 7-A ARM 7-R a c ec e f e , e e ac e-LOW e e e a , **nIRQ** a d **nFIQ**. H e e , GIC a c ec e ec f ca de c be e c f e e e e a , e ca a f e a c ec ed ce . T e ef e, de c be ac e-HIGH e e e , IRQ a d FIQ.

Interrupt bypass, GICv1 with GIC Security Extensions

W e a GIC e e a e , a CPU e face ca de e e ce e e , IRQ a d FIQ. I a a e e IRQ a G 1 e , b ca e e FIQ a G 0 e . I c a e e a , e CPU e face ca c de e a b a f b e a . F ca e, Tab e 2-1 a e 2-28 GICC_CTLR c e GIC e .

Table 2-1 Interrupt signal bypass behavior, GICv1 with Security Extensions

GICC_CTLR register bits			GIC interrupt outputs	
FIQEn	**EnableGrp0**	**EnableGrp1**	**IRQ request behavior**	**FIQ request behavior**
0	0	0	B a	B a
		1	D e b GIC CPU e face	B a
	1	0	D e b GIC CPU e face	B a
		1	D e b GIC CPU e face	B a
1	0	0	B a	B a
		1	D e b GIC CPU e face	B a
	1	0	B a	D e b GIC CPU e face
		1	D e b GIC CPU e face	D e b GIC CPU e face

F c a e e a , F e 2-3 e a f e G 0 a d G 1 e .

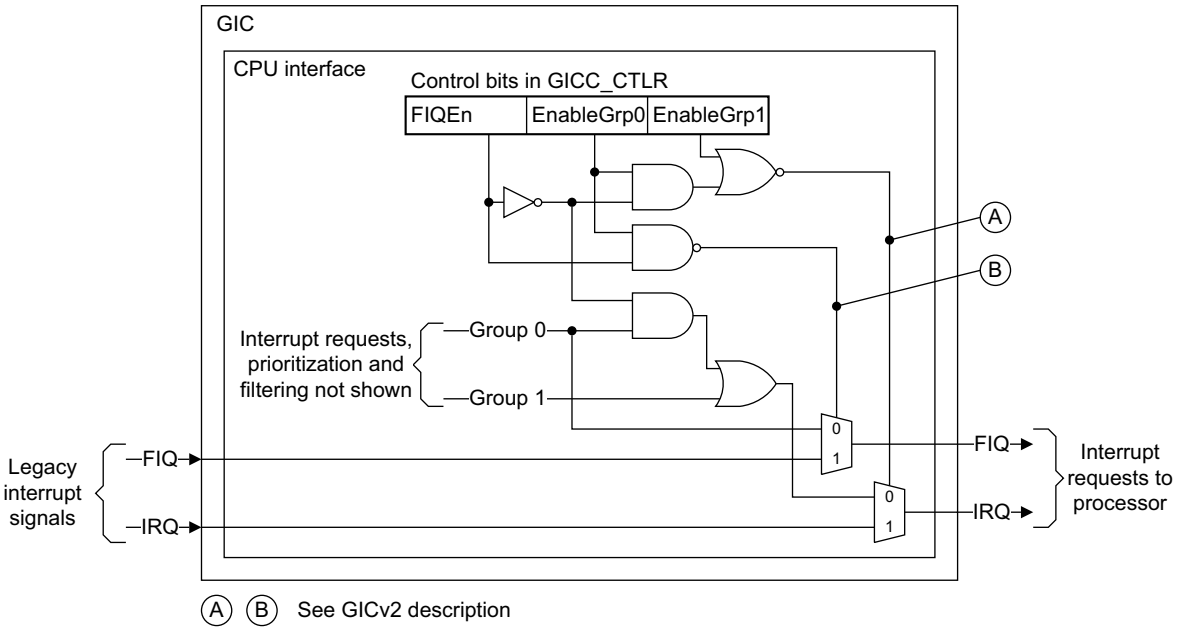


Figure 2-3 GICv1 Group 0 and Group 1 interrupt signaling, with interrupt signal bypass

GICv2 interrupt bypass, with bypass disable

W e a CPU e face a GIC 2 e e a c de e a b a , :

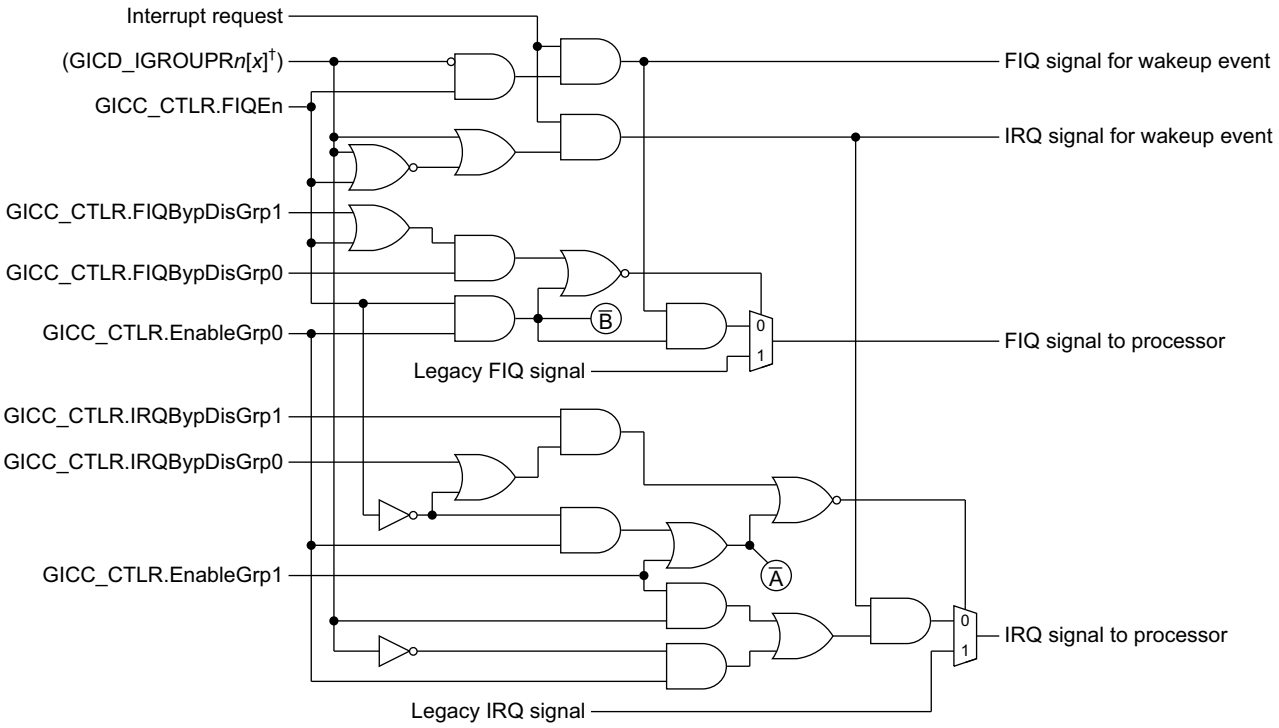
e e e b a c e e de c bed *I* , *GIC 1* *GIC S* *E* a e 2-27

add , e e GICC_CTLR c b a d ab e e e a b a f c a .

W e be d e b e CPU e face, eac e a ca be dea e ed a e a be d e b e e ac e . T be a c ed b e GICC_CTLR b a d ab e b :

FIQB D G 0
FIQB D G 1
IRQB D G 0
IRQB D G 1.

F e 2-4 e c c f e a f e b a CPU e face. *P , GIC 2* a e 2-31 e e f a ab e a e e e a F e 2-4.



† Values of *n* and *x* correspond to the requested interrupt
Ⓐ is the inverse of Ⓐ in the GICv1 implementation that supports interrupt grouping
Ⓑ is the inverse of Ⓑ in the GICv1 implementation that supports interrupt grouping

Figure 2-4 GICv2 interrupt bypass logic, with bypass disable

E a e 3-64 a de c be e a .

Tab e 2-2 a e 2-30 , e a CPU e face a a IRQ e e a c ec ed ce , b GICC_CTLR, a d e e e IRQ e e G 0 G 1, de e e e IRQ a b e e face. I e *IRQ* c f ab e:

Bypass I d ca e a e IRQ a e ce d e b e e ac IRQ a .

Deasserted I d ca e a e IRQ a e ce dea e ed.

Driven by GIC I d ca e a e IRQ a e ce d e b e GIC CPU e face c.

Table 2-2 IRQ request behavior, GICv2

GICC_CTLR register bits					IRQ for signaling	IRQ request signaling behavior
EnableGrp1	EnableGrp0	FIQEn	IRQBypDisGrp1	IRQBypDisGrp0		
0	0	0	0			B a
0	0	0	1			Dea e ed
0	0	1	0			B a
0	0	1	1	0		B a
0	0	1	1	1		Dea e ed
0	1	0			G 0	D e b GIC
0	1	0			G 1	Dea e ed
0	1	1	0			B a
0	1	1	1	0		B a
0	1	1	1	1		Dea e ed
1	0				G 0	Dea e ed
1	0				G 1	D e b GIC
1	1	0				D e b GIC
1	1	1			G 0	Dea e ed
1	1	1			G 1	D e b GIC

Tab e 2-3 , e a CPU e face a a FIQ e e a c ec ed ce , b GICC_CTLR e FIQ a b e e face:

Table 2-3 FIQ request behavior, GICv2

GICC_CTLR register bits				FIQ request signaling behavior
EnableGrp0	FIQEn	FIQBypDisGrp0	FIQBypDisGrp1	
0	0	0		B a , d e b e ac FIQ a
0	0	1	0	B a , d e b e ac FIQ a
0	0	1	1	FIQ e dea e ed
0	1	0		B a , d e b e ac FIQ a
0	1	1		FIQ e dea e ed
1	0	0		B a , d e b e ac FIQ a
1	0	1	0	B a , d e b e ac FIQ a
1	0	1	1	FIQ e dea e ed
1	1			D e b GIC CPU e face

2.3.2 Power management, GIC v2

T e GIC 2 a c ec e a e e e e a a e e a e e .

A F e 2-4 a e 2-29, e GIC 2 e b a c de c bed *GIC 2* , a e 2-28 c de a a ca be ed a a e a a e e c e . T e e a a e a a ab e e e e b e a b e GIC, a d e b a , a e d ab ed.

I add , e GICC_APR e e de f e e a d e a e e - a a e e a ca . H e e , e e a N - ec e acce e d e fe e Sec e e a , Sec e a d N - ec e c e f e e e e a e ded.

Interrupt Handling and Prioritization

T c a e de c be e e e e f e a d a d a e GIC. I c a e f ec :

3.1 About interrupt handling and prioritization

T e f b ec e e f a ab e e ed b a GIC, a d a c ec ed ce de e e a e f e ID ed b e GIC:

H a e 3-35

I a e 3-35.

T e e a de f e c a e de c be e a d a d a .

I e a d de c be :

e GIC ec e e

f a e ca a e GIC c f e a d c e

e a e ac e e GIC a a f eac e eac CPU e face

e e ce de f a ce e ac e GIC.

P a de c be :

e c f a a d c f e

e de f e ec f e d e

e de e a f e e a e b e a a e ce , c d :

G e a

G

G ee f a ac e e .

T e f ec de c be e a d a d a :

G a e 3-37

I a e 3-44.

T e GIC a c ec e ce a d ce e :

a ce e e GIC a a e ce e face, e *CP*

a ce e e GIC a a CPU e face f eac c ec ed ce .

I e e a ce a ce e , a GIC e e a ca c de e GIC Sec E e . T e GIC Sec E e :

ec e a a c ec ed ce a e e e ARM Sec E e a e e e Sec e acce e N - ec e acce e e GIC e e

e e e GIC e e a e acc f Sec e a d N - ec e acce e , a :

G e e e a e , de e a a e Sec e a d N - ec e c e

G e e e a e *S* , ea e a e acce b e Sec e acce e

G e e a e e a e *C* , ea e a e acce b e b Sec e a d N - ec e acce e .

e e GIC e fea e e a d f Sec e a d N - ec e e , c ca e:

G G 0 e a e Sec e e

G G 1 e a e N - ec e e .

a ce e , e e e GIC Sec E e e f CPU e face .

E ce f a GIC 1 e e a a d e c de e GIC Sec E e , a e e a f e GIC a c ec e . W e :

b defa , a e a e G 0 e , a d a e a ed a c ec ed ce e IRQ e e e

eac e ca be c f ed a G 1 e , a a G 0 e

a CPU e face ca be c f ed a G 0 e a c ec ed ce e FIQ e e e .

I e ,a d e GIC Sec E e , a e e a d a ec e . T e f ec de c be e effec f e a d e GIC Sec E e :

T a e 3-48

I a e 3-53.

3.1.1 Handling different interrupt types in a multiprocessor system

A GIC a d - , ee *I* a e 1-18.

I a ce e e a e GIC a d e :

f a e e e a ed e (SGI) e GIC N-N de

e e a (a d a e) e e GIC 1-N de .

See *M* a e 1-19 f def f e de .

3.1.2 Identifying the supported interrupts

T e GIC a c ec e def e d ffe e ID a e f e d ffe e e f e , ee *I ID* a e 2-24. H e e, e e e e e f e GIC e e a c b c f e ID f a e e.

Note

ARM ec e d a e e ed e a e ed e e e ID be a d a a a e f e ID a b e, beca e ed ce e be f e e a be e e ed, a d a d c e e c ec .

T c ec a d e e , f a e a e ID a e ed b e GIC. S f a e ca e e GICD_ISENABLER d c e f a . If e ce e e e ARM Sec E e , Sec e f a e d e e e e e a a e be N - ec e f a e. T e N - ec e f a e c e ca ee, a d e d c e ce f d f a .

GICD_ISENABLER0 de e Se -e ab e b f b :

SGI , e ID 15-0, c e d e e b [15:0]

PPI , e ID 31-16, c e d e e b [31:16].

T e e a GICD_ISENABLER , f GICD_ISENABLER1, de e Se -e ab e b f e SPI , a a e ID 32.

If a e :

ed, e Se -e ab e b c e d e ID RAZ/WI

ed a d e a e e ab ed, e Se -e ab e b c e d e ID RAO/WI.

S f a e d c e e e a a e ed b :

1. Read e GICD_TYPER. T e GICD_TYPER.ITL e N be f e d de f e e be f e e ed GICD_ISENABLER , a d e ef e e a be f SPI a be ed.
2. W e GICD_CTLR d ab e f a d f e f e d b e CPU e face . F e f a , ee *E D CP* a e 4-77.
3. F eac e e ed GICD_ISENABLER , a GICD_ISENABLER0:
   W `0xFFFFFFFF` e GICD_ISENABLER .
   Read e a e f e GICD_ISENABLER . B a ead a 1 c e d ed e ID .

S f a e e e GICD_ICENABLER d c e e e a a e e a e e ab ed. F eac e e ed GICD_ICENABLER , a GICD_ICENABLER0, f a e:

1. W e 0xFFFFFFFF e GICD_ICENABLER . T d ab e a e a ca be d ab ed.
2. Read e a e f e GICD_ICENABLER . B a ead a 1 c e d e a a e e a e e ab ed.
3. W e 1 a GICD_ISENABLER b c e d e a be e-e ab ed.

T e GIC e e e a e be f GICD_ISENABLER a d GICD_ICENABLER .

W e f a e a c e ed d c e , ca e e GICD_CTLR e-e ab e f a d f e f e D b e CPU e face .

If e GIC e e e GIC Sec E e , f a e ca e Sec e acce e d c e a e ed e ID , ee *T* a e 3-48 f e f a .

S f a e N - ec e acce e ca d c e a d c e e a a e c f ed a N - ec e.

If Sec e f a e c a e e ec c f a fa e af e N - ec e f a e a d c e ed ed e , c ca e e effec f e c a e e N - ec e f a e.

I a GIC a de e , f a e ca :

e e GICD_IGROUPR e e , c f e e a G 0 G 1

c e f a d f G 0 a d G 1 e de e de , e GICD_CTLR.E ab eG 0 a d GICD_CTLR.E ab eG 1 b .

3.2 General handling of interrupts

T e D b a a a a e ac e f eac ed e eac CPU e face. *I* a e 3-41 de c be a e ac ea d a e a . T e b e a e f a e a e:

ac e
e d
ac e
ac e a d e d .

Note

T ec e a e e f e a d f e a GIC e e a a d e c de e GIC Sec E e . I d e e a f de c f a d ed e . I e , a d e GIC Sec E e , e e d e ba c de f GIC e a de c bed ec . F e f a ee *T* a e 3-48.

T ba c de f e a d a a e e a d f a e a e e a a c de e GIC V a a E e . F e f a , ee C a e 5 *GIC S* .

W e e GIC ec e a e e e , a a e a a . Re e e a a e d e d e affec e a e f e e .

T e GIC e a d e e ce :

1. T e GIC de e e e e a a e e ab ed.
2. F eac e d e , e GIC de e e e a e ed ce ce .
3. F eac CPU e face, e D b f a d e e e d e a a e a e face.
4. Eac CPU e face de e e e e e a a e e e ce , a d f e ed, d e .
5. T e ce ac ed e e e , a d e GIC e e e ID a d da e e e a e.
6. Af e ce e e , e ce a *E I* (EOI) e GIC.

I e de a , e e e a e a f :

1. T e GIC de e e e e e eac e e ab ed. A e a e ab ed a effec e GIC.
2. F eac e ab ed e a e d , e D b de e e e a e ed ce ce .
3. F eac ce , e D b de e e e e e d e , ba ed e f a d f eac e , a d f a d e e e a e ed CPU e face .
4. If e d b f a d a e e e e a CPU e face, e CPU e face de e e e e e e a *S* be a ed e ce . If e e a ff c e , e GIC a a e e e e ce .
5. W e a ce a e e e e ce , ead e GICC_IAR f CPU e face ac ed e e e e . T ead e a I e ID, a d f a SGI, e ce ce ID, a e ce e e ec e c ec e a d e . W e ec e ead, e GIC c a e e a e f e e a f :

   f e e d a e f e e e e e e e bec e ac e, f e e e e a ed a a , f e d ac e a d e d .

   e e, f e d ac e

Note

A e e - e e e a e e ac ed ed b e ce , beca e e e a e GIC e a a e ed e *I S R* (ISR) e ce acce e e e a a e e a .

I a ce e e a , e GIC a d e :

- G PPI a d SGI e GIC N-N de , e e e ac ed e e f a e b e ce a effec e a e f e e e CPU e face
- G SPI e GIC 1-N de , e e e ac ed e e f a e b e ce e e e e d a f e e a e a e ed ce , ee *I I-N* a e 3-41.

I GIC 2, e a f a e de e GICC_CTLR.Ac C b e 0, e a a e e e a e ed a a e G 0 a d G 1 e , a f :

- G GICC_IAR, GICC_EOIR, a d GICC_HPPIR f G 0 e
- G GICC_AIAR, GICC_AEOIR, a d GICC_AHPPIR f G 1 e .

ARM de eca e e e f GICC_CTLR.Ac C , a d ec e d a f a e de e e GICC_CTLR.Ac C e 0, ee *T* a e 3-50.

6. W e e ce a c e ed a d e e , a c e e GIC. A de c bed *P* , :

   a a e e a a d e a (EOIR)

   a e e a b e e e e deac a e e e e , GICC_DIR.

   F eac CPU e face, e GIC a c ec e e e e de f e a d e a EOIR be e e e e f e de f e ead f e GICC_IAR GICC_AIAR, a eac a d EOIR e efe e ece e ac ed e.

   If, af e e EOIR e, e e e d e f *S* , e CPU e face dea e e e ce e e e ce .

   A CPU e face e e a e c ec ed ce a e a ac e a d e d . I a e a a e e d a d a e ff c e :

   - F PPI a d SGI , e ac e a f a c a e ID ba ed be ee CPU e face . T ea a f a a c a e ID ac e ac e a d e d a CPU e face, e e a a e ID a ed a CPU e face.
   - F SPI , e ac e a f a e c a CPU e face . T ea a f a e ac e ac e a d e d e CPU e face e a ed a CPU e face.

F e f a ab e e ce ee:

P
I a e 3-44
T a e 3-48
I a e 3-53.

3.2.1 Priority drop and interrupt deactivation

W e a ce c e e e ce f a e , a c e e GIC. I e c e e e e f c a e e GIC a e:

Priority drop P d e d e *R* a cc a a d e a EOIR, e e e GICC_EOIR e GICC_AEOIR. A a d e a e a UNPREDICTABLE, ed, a d a e ID a e ea e a 1019.

O d , e ed ced f e f e e efe e ced b e EOIR e e e :

e f e e - ac e e f c e e a bee EOIR e

e *I* , f e e ac e e f c e e a bee EOIR e.
See *P* a e 3-45 f e f a ab .

Interrupt deactivation

I e deac a e c a e f e a e fa e , e e :
f ac e a d e d , e d
f ac e, d e.

O a GIC 1 e e a , a d a GIC 2 e e a e GICC_CTLR.EOI de e 0, a a d EOIR e a deac a e e e efe e ce .

O a GIC 2 e e a , e GICC_CTLR.EOI de 1 e a a e e d a d e deac a e a , a d e a d f a e :

1. Pe f a a d EOIR e, ca e d e GIC CPU e face.
2. S b e e , e e GICC_DIR, deac a e e e .

T e GIC a c ec e ec f ca e e a a d EOIR e a e de ed, a :
a a d GICC_EOIR e c e d e ece ac ed ed e
a a d GICC_AEOIR e c e d e ece ac ed ed G 1 e .
e e a GICC_EOIR e affec G 0 G 1 e de e d b :
G e a e f e GICC_CTLR. Ac C b
G f e GIC e e e GIC Sec E e , e e e e Sec e N - ec e.

Note

I a GIC 2 e e a a c de e Sec E e :
GICC_AEOIR a a a f e N - ec e c f GICC_EOIR
GICC_AIAR a a a f e N - ec e c f GICC_IAR
GICC_AIAR a d GICC_AEOIR a e Sec e e e , ea e a e acce b e b Sec e acce e .

T e e de e e e f GICC_DIR e . H e e , e effec UNPREDICTABLE f f a e e GICC_DIR e :

GICC_CTLR.EOI de e 0

GICC_CTLR.EOI de e 1 a d e e a bee a c e d e GICC_EOIR GICC_AEOIR.

W e a ca e , ARM ec e d a , f eac CPU e face a c e d a ce a ac e :
GICC_CTLR.EOI de b e 1
f e GIC e e e GIC Sec E e , e GICC_CTLR.EOI deNS b e 1

See *C* a e 5-161 f e f a .

3.2.2 Interrupt controls in the GIC

T e f ec de c be e e c e GIC:

I
S a e 3-40
F a e 3-40
G *SGI* a e 3-40.

Interrupt enables

F e e a e , a ce :
e ab e a e b e a a e GICD_ISENABLER b
d ab e a e b e a a e GICD_ICENABLER b .

W e e SGI a e e a e e ab ed, ca be e ab ed a d d ab ed b e e GICD_ISENABLER a d GICD_ICENABLER , IMPLEMENTATION DEFINED.

W e e GICD_ISENABLER a d GICD_ICENABLER c e e e D b f a d ec f c e e CPU e face . D ab a e b e a a e GICD_ICENABLER d e e e a e f c a a e, f e a e bec e d .

Setting and clearing pending state of an interrupt

F e e a e , a ce ca :

e e e d a e b e a a e GICD_ISPENDR b

c ea e e d a e b e a a e GICD_ICPENDR b .

F a e e - e e e :

If e ad a e a f a e a e ed e a ce e ec e d GICD_ICPENDR b e e e e e a effec e e d a e f e e .

If a ce e a l a GICD_ISPENDR b e ec e d e bec e e d e a d e f e a e f e a d a e a f a e , a d e a e d e a d e f e a e dea e f e a .

F e f a ab e c f e e d a e fa e e - e e ee *C* - a e 4-100.

F SGI , e GIC e e ec e d GICD_ISPENDR a d GICD_ICPENDR b . A ce ca c a e e a e fa f a e - e e a ed e b e e e e . T ca , a SGI ade e d b e GICD_SGIR. I GIC 2, e e d a e f SGI ca a be d f ed d ec e GICD_SPENDSGIR a d GICD_CPENDSGIR b .

Finding the active or pending state of an interrupt

A ce ca f d:

e e d a e fa e b ead e c e d GICD_ISPENDR GICD_ICPENDR b

e ac e a e fa e b ead e c e d GICD_ISACTIVER GICD_ICACTIVER b .

T ec e d e e b 1 f e e e d ac e. If a e e d a d ac e e c e d b 1 b e e .

W e e e e GIC a e, a ce a e acc f e e d a d ac e a e fa e . F e f a ee *P GIC* a e 4-155.

F a SGI, ec e d GICD_ISPENDR a d GICD_ICPENDR b RAO f e e a e d e f a ea e e e a ce a a e e ce ead e GICD_ISPENDR GICD_ICPENDR . I GIC 2, e ce a e e SGI ca a be de e ed b ead e c e d GICD_SPENDSGIR GICD_CPENDSGIR b .

Generating an SGI

A ce e e a e a SGI b a GICD_SGIR. A SGI ca a e e ce , a d e GICD_SGIR e ec f e e a e ce . T e GICD_SGIR c de a f :

e e ce a e e GICD_SGIR

e a ce e a e e a e e GICD_SGIR.

SGI f d ffe e ce e e a e e ID . T e ef e, a a e ce ca ece e SGI e a e e ID f d ffe e ce . H e e, e e d a e fa SGI a e de e de f a f e f a e d ffe e :

e ID

ce ce

a e ce .

O e e a ec f c e ID ca be ac e a CPU e face a a e. T ea a a CPU e face ca a e SGI e a e e ID ac ea e a e e, e e f d ffe e ce a e a ed SGI e a e e ID a ce .

O e CPU e face f e a e ce , ead e GICC_IAR f a SGI e b e e ID a d e CPU ID f e ce a e e a ed e e , e f e e . T ec b a f e ID a d ce CPU ID e de f e e e e a e ce .

I a ce e e a , e e f eac SGI e ID def ed de e de f eac a e ce , ee *I P R , GICD IPRIORIT R* a e 4-104. F eac CPU e face, a SGI a a c a e ID a a e e d a e face a e e a e a d be a d ed e a . T e de c e CPU e face e a e e e SGI IMPLEMENTATION SPECIFIC.

3.2.3 Implications of the 1-N model

I a ce e e a , e GIC e e GIC 1-N de , de c bed *M* a e 1-19, a d e e e a e a a e e a e ce , a , SPI . T ea a e e GIC ec e a e ac ed e f e f e a e ce c ea e e d a e f e e a e e a e ed ce . A GIC e e a e e a a e be a d ed e 1-N de ac ed ed b e CPU e face, a d a a e e face e a e ID.

W e e a e ce a e ac ed e e e , e f ca cc :

A ce ead e GICC_IAR a d b a e e ID f e e be e ced.

Note

I GIC 1, e a e a e ce a e b a ed e ID, f e ce ead e GICC_IAR e e a e a e . T e e e e f a e e a e ce e e a e ce e e ce e. A ca ec a ac e e e e , a ed e , a c e (ISR).

A ce ead e GICC_IAR a d b a e e ID 1023, d ca a e . T e ce ca e f e e ce e GICC_EOIR.

T e e ID d ca e a e a e e e d , ca beca e a e a e ce a d .

Note

A GIC 1 e e a e e a e ce ca a e a 1-N e ac e, e e e e e f a c e ISR. T e ed b e a c ec e, a d e e c GIC c de e be a .

F a ce , f a e ac e a d e d , e GIC d e a a e e ce e e f e e a ce e ac e a c ea ed.

3.2.4 Interrupt handling state machine

T e GIC a a a a e ac e f eac ed e eac CPU e face. F e 3-1 a e 3-42 a a ce f a e ac e, a d e be a e a .

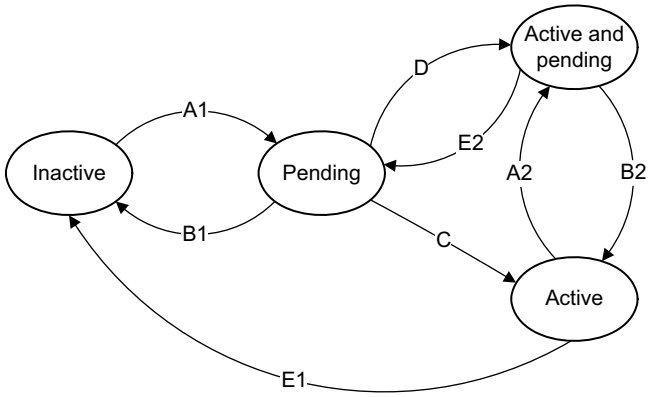


Figure 3-1 Interrupt handling state machine

Note

SGI a e e a ed b e GICD_SGIR GICD_SPENDSGIR . Pe e a e a e e a ed b e e e a e fa a d a e e e e a e GIC, b a e a GICD_ISPENDR .

A de c bed *P* a e 3-38:

G a GIC 1 e e a , d a a a c a ed e deac a

G a GIC 2 e e a , d ca be e a a ed f e deac a .

F e 3-1 d e b e e a a f d a d e deac a . T a e e Ac e a e.

W e e f a d b e D b a d e a b e CPU e face a e e ab ed, e c d a ca e eac f e a e a a e a f :

Transition A1 or A2, add pending state

F a SGI, cc f e e :

S f a e e a GICD_SGIR a ec f e e ce a a a e .

S f a e e a e ce e e GICD_SPENDSGIR b a c e d e e ed ce ce a d e ID

Note

If e GIC e e e GIC Sec E e a d e e e GICD_SGIR Sec e, e a cc f e ec c f a f e ec f ed SGI, f e a a e CPU e face, c e d e GICD_SGIR.NSATT b a e.

F a SPI PPI, cc f e e :

a e e a a e a e e e a f a e e a GICD_ISPENDR .

Transition B1 or B2, remove pending state

F a SGI, cc f f a e e a e ce e e e e a b f e GICD_CPENDSGIR .

F a SPI PPI, cc f e e :

e e e - e e e d beca e f e a e fa a , a d a a dea e ed

e e e d beca e f e a e fa ed e- e ed e a , a e a GICD_ISPENDR , a d f a e e ec e d GICD_ICPENDR .

Transition C, pending to active

If e e e ab ed a d f S ff c e be a ed e ce , cc e f a e ead f e GICC_IAR.

Transition D, pending to active and pending

F a SGI, a cc e e f ef c c a ce :

If a e e e SGI a e e d cc a a a e e a e e a a ead f GICC_IAR.

W e e e d SGI e a e e ID a e f e a e ce ce a d a e e a e ce . If e f e SGI f a C, e e SGI f a D

F a SPI PPI a cc f a e f a :

T e e e ab ed.

S f a e ead f e GICC_IAR. T ead add e ac e a e e e .

I add , e f e f c d a e :

G F a e e - e e e , e e a e a a e ed. T a e ca e, beca e e e e a d e dea e e e e ce a e ced e e .

G F a ed e- e ed e , e e a cc de e d e f e ead f e GICC_IAR e a e e de ec f e ea e f e e . O e e e ead f e GICC_IAR ca e a C, b f ed b a A2.

Transition E1 or E2, remove active state

Occ e f a e deac a e a e b e e GICC_EOIR GICC_DIR. F e f a ee *P* a e 3-38. I a GIC e e a e c de e V a a E e , a cc f e a CPU e face a a e c e d ca e a bee deac a ed.

3.2.5 Special interrupt numbers

T e GIC a c ec e e e e e ID be 1020-1023 f ec a e . I a GIC 1 e e a a d e e e GIC Sec E e , e e f e e ed ID 1023. T a e e ed a ce , e e a e ac ed e, f e e e d e ff c e f be a ed e ce . I de c bed a a e e a .

Note

A ace c d ca ca e a e . F e a e, a e ca cc f a ce e a 1 a f e d a GICD_ICENABLER a c e d a e d e af e e CPU e face a a ed e e e ce a d e ce a ec ed e e , b bef e e ce a ead f e GICC_IAR.

F e f a ab e ec a e be ee *S GIC* a e 3-50.

3.3 Interrupt prioritization

T ec de c be e a e GIC a c ec e. I c de e f b ec :
P a e 3-45
P a e 3-45
P a e 3-45
I a e 3-47.

Note

T ec de c be a e e a a e , a d d e c de e GIC Sec E e . I e , a d e GIC Sec E e , e e d ba c de f GIC e a . F e f a , ee *I* a e 3-53.

S f a ec f e e a e GIC b a a a e eac e ce. P a e a e 8-b ed b a . A GIC a f 16 a d a a f 256 e e . If e GIC e e fe e a 256 e e , - de b f e f e d a e RAZ/WI. T ea a e be f e e ed f e d b IMPLEMENTATION DEFINED e a e 4-8, a Tab e 3-1 .

Table 3-1 Effect of not implementing some priority field bits

Implemented priority bits	Possible priority field values	Number of priority levels
[7:0]	`0x00-0xFF` (0-255), a a e	256
[7:1]	`0x00-0xFE`, (0-254), e e a e	128
[7:2]	`0x00-0xFC` (0-252), e f 4	64
[7:3]	`0x00-0xF8` (0-248), e f 8	32
[7:4]	`0x00-0xF0` (0-240), e f 16	16

I e GIC a c e e, e be a e e , a , e e ea ed a e e e e f e e . P f e d a e 0 a a d ca e e e be e , a d e e a e de e d e be f e e ed e e , a Tab e 3-1 .

T e GICD_IPRIORITYR e e d e a e f eac ed e . A e e a e e ea e f a a c a e a da a f ed a e , ea e a e f a e ead- . F e e , f a e e e GICD_IPRIORITYR e e e e e e . I IMPLEMENTATION DEFINED e e a e GICD_IPRIORITYR c a e e f a ac e e .

T de e e e be f b e e ed, f a e ca e `0xFF` a ab e GICD_IPRIORITYR f e d, a d ead bac e a e ed.

Note

ARM ec e d a , bef e c ec e a e a :
f a e ea e , f a e f d ab e e e f a SGI, f a e f c ec a e e ac e.

If, a a c a CPU e face, e e d e a e e a e , a d a e *S* f e e face a e e ce , IMPLEMENTATION SPECIFIC e e face e ec c e a .

W e a e ac e a CPU e face, e GIC a a e - e a CPU e face, ee *P* a e 3-45.

3.3.1 Preemption

A CPU e face a f e e d e a a e ce bef e a ac e e c e e e . A e d e a ed f b :

I e a e a f a CPU e face, ee *P* .

I e a a f e *R* e CPU e face, ee *P* a d *R P R* , *GICC RPR* a e 4-142.

P ee cc a e e e e ce ac ed e e e e , a d a e ce efe e ce e e ac e e e c e ce . W e cc , e a ac e e a d a e bee . S a e ce a e e a e e ac e e e de c bed a .

Note

F a ce a c e e ARM a c ec e:

G T e a e f e I F b e CPSR de e e e e e ce e d e a ed e b a e e ac ed e ced e.

G W e ce a ee e , e ce a e a d a e e e ec e f e e ac e ISR.

F e f a , ee e *ARM A R M* , *ARM 7-A ARM 7-R* .

P ea e f a e e affec e *R* e CPU e face, a d e ef e d e e e e ee . I GIC 1 e e a , d a e e a e deac a ed, b GIC 2 e e a , d a d e deac a ca be e a a ed. F e f a ee *P* a e 3-38.

3.3.2 Priority masking

T e GICC_PMR f a CPU e face def e a e d f e a e ce . T e GIC a e d e a e a e d a e e a e ce . A a e f e , e e e e a e, a a e f be a ed e a c a e d ce . T e GIC d e e c a e fa e d e e e d.

T e GIC a a a a e a a e a e ed f e d a e. T de a add a ea f e e a e be a ed a ce .

Note

W 255 e GICC_PMR a a e e a e ed f e d a e. Tab e 3-1 a e 3-44 e a e ed f e d a e a e e be f e e ed b .

3.3.3 Priority grouping

P e e B a P Re e , GICC_BPR, a a e f e d , e a d e . W e de e ee , a e e a e a e c de e d a e e a , e a d e f e b . T ea a e e ca be e e ac e a eac . T e ac e a a e P ee e e . F e f a , ee *A P R* , *GICC APR* a e 4-149.

T e GIC e e f e d de e e e e a e d e a ff c e ee a ac e e , a f :

F a e d e ee a ac e e , be e a e f eac e e . T a , e a e f e f e d f e e e be e a e a e f e f e d f e *R* .

If e e a e ac e e e CPU e face, e e e d e ca be a ed a ce , e a d e f e .

I eac ca e, e e d e c a ed e a , a d e e a ed f a ed. F e f a , ee *P* a e 3-45.

T e b a f e d e GICC_BPR c e f e b e a . T 3-b f e d ec f e a f e ea f ca b f e 8-b e f e d a e e c ded f e f e d, a Tab e 3-2 .

Table 3-2 Priority grouping by binary point

Binary point value	Interrupt priority field [7:0]		
	Group priority field	Subpriority field	Field with binary point
0	[7:1]	[0]	.
1	[7:2]	[1:0]	.
2	[7:3]	[2:0]	.
3	[7:4]	[3:0]	.
4	[7:5]	[4:0]	.
5	[7:6]	[5:0]	.
6	[7]	[6:0]	.
7	N ee	[7:0]	.

T e b a a e ed IMPLEMENTATION DEFINED e a e 0-3.

GIC 1 e e a e GIC Sec E e a d GIC 2 e e a a e b a e e . T e c f e b a e e ed ca c a e de e d e e e e a G 0 e a G 1 e , a def ed b e GICD_IGROUPR e e , a d a e a e f e GICC_CTLR.CBPR b .

Tab e 3-2 c b a e e ed f d ffe e GIC e e a .

Table 3-3 Binary point register used to calculate priority grouping

GIC implementation	Condition	
	(Group 0 interrupt) || CBPR==1[a]	(Group 1 interrupt) && CBPR==0
GIC 1 Sec E e [b]	-	-
GIC 2 Sec E e	GICC_BPR	GICC_ABPR
GIC Sec E e	Sec e GICC_BPR	N - ec e GICC_BPR[c]

a. GICC_CTLR.CBPR. N e e ed a GIC 1 e e a a d e c de e Sec E e .
b. A GIC 1 e e a Sec E e a e a d e b a e e , GICC_BPR, a a a e de e e e .
c. F a GIC 2 Sec E e , e GICC_ABPR a d N - ec e GICC_BPR a e a a e f e a e e e .

W e e GICC_CTLR.CBPR b e 1, f a e ca c f e e CPU e face de e e e f a G 1 e e a e b a e e a f a G 0 e .

W e e e e d e a e e a e , e GIC e e b f e d e e e a . W e e e e d e a a e e a e b , e GIC e ec be ee e e IMPLEMENTATION SPECIFIC.

3.3.4 Interrupt generation

T e e d c de *E* a e 3-64 de c be e e e a f e b e GIC.

3.4 The effect of interrupt grouping on interrupt handling

T ec de c be e effec f e a d e GIC Sec E e e a d .

A GIC 1 e e a a c de e GIC Sec E e , a GIC 2 e e a , de e a f IRQ a d FIQ e ce e e :

T e CPU e face a a e e IRQ e ce e e f G 1 e
S f a eca c f e e CPU e face e e e IRQ FIQ e ce e e f G 0 e .

A e- , af e a e e, a GIC e e a c f ed e a e e a, a de c bed *GIC* a e 3-51.

T e e a de f ec de c be a GIC a e e e , a f :

GIC
S *GIC* a e 3-50
T a e 3-50
GIC a e 3-51.

3.4.1 GIC interrupt grouping support

Note

I a GIC 1 e e a , e ded a a f e GIC Sec E e .

T e GICD_IGROUPR e e c f e eac e a G 0 G 1.

I a CPU e face, a GIC 2 e e a , e GICC_* a a e e ca de de e de c f G 0 a d G 1 e e , a Tab e 3-4 .

Table 3-4 CPU interface control of Group 0 and Group 1 interrupts, GICv2

Function	Register, Group 0	Register, Group 1
B a [a]	GICC_BPR	GICC_ABPR
I e ac ed e	GICC_IAR	GICC_AIAR
EOI	GICC_EOIR	GICC_AEOIR
H e e d e	GICC_HPPIR	GICC_AHPPIR

a. See Tab e 3-3 a e 3-46 f e f a .

I a e e a a c de e GIC Sec E e , ea a e e :

ca e e e a a e f e N - ec e c f e G 0 e e , f e a e GICC_ABPR a a a f e N -Sec e c f GICC_BPR

a e acce b e b Sec e acce e .

I a GIC 1 e e a a c de e GIC Sec E e :

T e e e ed a a e e GICC_ABPR

T e e c f e G 1 e a e ded b e N - ec ec e f e G 0 c e e , a Tab e 3-5 .

Table 3-5 CPU interface Non-secure control of Group 1 interrupts

Function	Non-secure Group 1 control register
B a	N - ec e GICC_BPR
I e ac ed e	N - ec e GICC_IAR
EOI	N - ec e GICC_EOIR
H e e d e	N - ec e GICC_HPPIR

I a GIC e e a a c de e GIC Sec E e , CPU e face N - ec e c f G 1 e de ca GIC 1 a d GIC 2. T ea a , a GIC 2 e e a , Tab e 3-5 e GICC_* e e a de e N - ec e c f G 1 e .

I a e e a a e , GICC_CTLR c a add a f e d , c d f e d c e a d f e ed e :

Se a a e e ab e b c e a f G 0 a d G 1 e e c ec ed ce :

G b [0], e E ab e b a GIC a d e e , bec e e E ab eG 0 b , a d c e e G 0 e a e a ed e ce

G e E ab eG 1 b added, c e e G 1 e a e a ed e ce .

T e FIQE b , a c e e e face a G 0 e e ce e IRQ FIQ e e e .

T e CBPR b , a c e e GICC_BPR GICC_ABPR ed e de e b e e ee b G 1 e , ee *C G I* a e 3-57.

T e Ac C b , a c e e a ead f e GICC_IAR, e Sec e GICC_IAR f e GIC e e e Sec E e , ca ac ed e a G 1 e . F e f a ee *T* a e 3-50.

Note

A de c bed *T* a e 3-50, ARM de eca e e e Ac C b 1.

I a GIC 2 e e a :

G e IRQ a d FIQ b a d ab e b , a c e e e b a IRQ a d FIQ a a e f a ded e ce , ee *I , GIC 2* a e 2-27.

G T e EOI de b , a c e e d e a a ed f e deac a , ee *P* a e 3-38. If e GIC e e e Sec E e , e a a e EOI deNS a d EOI deS b a e e e ed f N - ec e a d Sec e acce e . T de de e de c f e E d f e de f N - ec e a d Sec e e a d .

3.4.2 Special interrupt numbers when a GIC supports interrupt grouping

S a e 3-43 de c be e e f e ID 1023 d ca e a . T e f f e e ID be e GIC a c ec e e e e f ec a e a f :

1020-1021 Re e ed.

1022 U ed f e GIC e .

T e GIC e a e a ce e e a e ac ed e e a f e f a :

e e ac ed e a ead f GICC_IAR

e e e d e a G 1 e GICC_CTLR.Ac C e 0

e f e e ff c e f be a ed e ce .

Note

I e ID 1022 d ca e a e e a G 1 e f ff c e be a ed e ce , a be ac ed ed b a ead f e GICC_AIAR, a e e a a c de e GIC Sec E e , b a ead f e N - ec e GICC_IAR.

W e a GIC 1 e e a , a Sec e f a e a ce a e c ed e e N - ec e f a e a de e e , e e e a e c .

1023 T a e e ed a ce , e e a e ac ed e, f e e d e ff c e f be a ed e ce .

O a ce a e , a e f 1022 a d 1023 a e e ID .

3.4.3 The effect of interrupt grouping on interrupt acknowledgement

I a GIC e e a a d e e , e a ce a e a e , ac ed e e e b ead e GICC_IAR, ee *G* a e 3-37. T ead f e GICC_IAR a a ac ed e e e e d e f e ce e f e ead.

I a GIC e e a a e , ARM ec e d e GICC_CTLR.Ac C 0, ea :

f a GIC 2 e e a :

G a 0 e ac ed ed b a ead f GICC_IAR, a Sec e ead f GICC_IAR f e e e a c de e GIC Sec E e

G a 1 e ac ed ed b a ead f GICC_AIAR, a N - ec e ead f GICC_IAR f e e e a c de e GIC Sec E e

f a GIC 1 e e a :

G a 0 e be ac ed ed b a ead f e Sec e GICC_IAR

G a 1 e be ac ed ed b a ead f N - ec e GICC_IAR.

I eac ca e, e ead be a ac ed e e f e e e d e e CPU e face.

F e f a ab e e e ed f e a d , ee *GIC* a e 3-48.

If e I e Ac ed e e e acce d e c e d e e - e d e e CPU e face e :

a ead f GICC_IAR e e e - e d e a G 1 e e e e a e 1022

a ead f GICC_AIAR e e e - e d e a G 0 e e e e a e 1023.

W e e GICC_CTLR.Ac C b e 0, e e e c ec e , e e G 0 e a e a e a a G 1 e .

W e e GICC_CTLR.Ac C b e 1, a ead f GICC_IAR ac ed e e e - e d e e CPU e face, e a d e f e e a G 0 a G 1 e . H e e , ARM de eca e e f GICC_CTLR.Ac C , a d ec e d a f a e de e e GICC_CTLR.Ac C e 0.

Interrupt acknowledgement with the GIC Security Extensions

T b ec de c be e e e e f ac ed ed e a e a d e a ce a e e e ARM ce Sec E e c ec ed a GIC CPU e face a c ded e GIC Sec E e . I c f a :

G 0 e a e Sec e e

G 1 e a e N - ec e e .

T e b ec de c be e a GICC_CTLR.Ac C e 0, e ec e ded c f a .

If e e e d e a Sec e e , e ce a e a Sec e ead f e GICC_IAR ac ed e .

T ac ed e a N - ec e e , e ce ca :

e f a N - ec e ead f e GICC_IAR e e

a GIC 2 e e a , e f a Sec e ead f e GICC_AIAR e e .

T ea a , e N - ec e f a e a d a N - ec e e , e ce a e a N - ec e ead f e GICC_IAR ac ed e a N - ec e e .

If a ead f e GICC_IAR d e a c e ec f e e , e GICC_IAR ead d e ac ed e a e a d e e a e:

1022 f a Sec e ead e e e e N - ec e

1023 f a N - ec e ead e e e e Sec e.

See *E* *GICC IAR* a e 4-136 f e f a .

3.4.4 GIC power on or reset configuration

O e - , af e a e e , a GIC e e a a e c f ed :

a e a ed G 0

e FIQ e ce e e d ab ed.

T ea a G 0 e a e a ed e IRQ e e e . F e 3-2 a e 3-52 c f a .

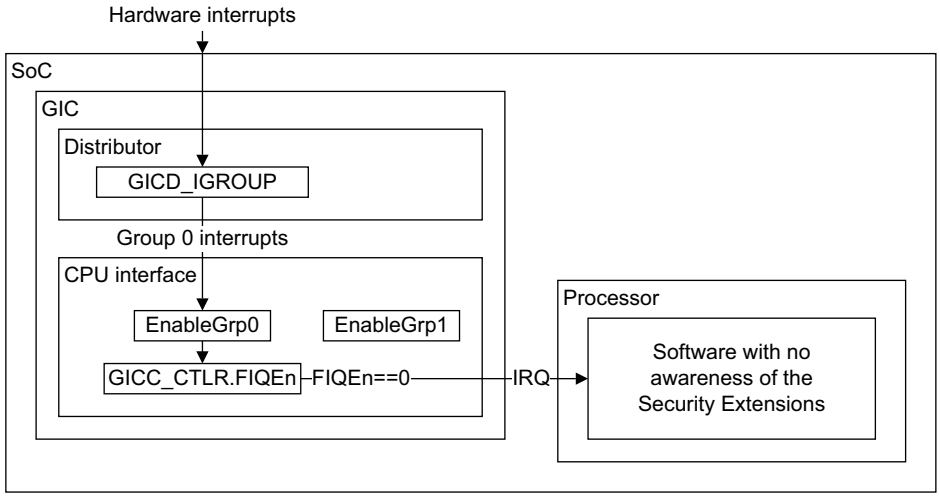


Figure 3-2 Reset configuration of a GIC that includes the FIQ exception request

3.5 Interrupt grouping and interrupt prioritization

Ma e e ea e ea G 1 e e ee a G 0 e . F c a e e a , ARM ec e d a:

G 0 e a ea a a ed a e e e af f e ed a e a e. T e e a e c e d e e- e

G 1 e a ea a a ed a e e e af f e ed a e a e. T e e a e c e d e e- e .

T e e a e e G 1 e f e a a G 0 e .

If e GIC e GIC Sec E e :

T e GIC de Sec e a d N - ec e e f e e e , ee *S GIC S E* .

T e be f a e ed c ea e f 16 32.

N - ec e acce e ca ee a f f e ed a e . T e ef e, f e GIC e e 32 a e , N - ec e acce e ee 16 a e .

Note

See *P S N - GIC* a e 1-20 f e def f Sec e f a e a d Sec e a d N - ec e acce e .

3.5.1 Software views of interrupt priority in a GIC that includes the Security Extensions

W e a ce ead e a e fa G 1 e , e GIC e e e e Sec e e N - ec e e f a a e, de e d e e e acce Sec e N - ec e. T ec de c be e e f e , a d e e a be ee e .

T e GIC e e a f32 a d a a f256 e e . T ea e e 5-8 b f e 8-b a e f e d e GICD_IPRIORITYR e e . A f e e e ed b ca be acce ed b a Sec e acce , a d e e ed - de b f e f e d a e RAZ/WI. F e 3-3 e Sec e e fa a e f e d f a e . T e a e ed e D b e a e e Sec e e .

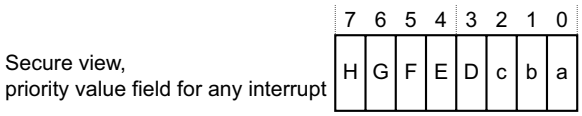


Figure 3-3 Secure view of the priority field for any interrupt

I e :

b H-D a e e b a e GIC e e , c e d 32 e e

b c-a a e e b e GIC e e , a a e RAZ/WI f e e ed.

e GIC e e e b H-a de e a 256 e e

ARM ec e d a , f a G 1 e , b [7] e 1.

A N - ec e acce ca ee a a e f e d a c e d e N - ec e e f e .

F N - ec e acce e , e GIC a f e e e f Sec e acce e . F e 3-4 a e 3-54 e N - ec e e fa a e f e d f a G 1 e .

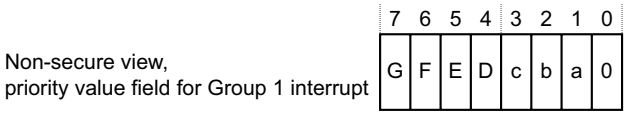


Figure 3-4 Non-secure view of the priority field for a Group 1 interrupt

I e :

b G-D a e e b a e GIC e e , c e d 16 e e

b c-a a e e b e GIC e e , a a e RAZ/WI f e e ed

e GIC e e b G-a de e a 128 e e

b [0] RAZ/WI.

T e N - ec e e f a a e d e e a e ed e D b . Ta e a e f a N - ec e e a f e d, bef e e a e:

e a e - f ed b e b

b [7] f e a e e 1.

T a a ea e a e f e G 1 e e a f f e be a e a e, ea e e e eb a f f e a e.

A Sec e ead f e a e f a e e e a e ed e D b . F e 3-5 Sec e e f e a e f e d f a G 1 e a a ad a e f e d e b a N - ec e acce , a ad a a e b [7] == 1 e b a Sec e acce :

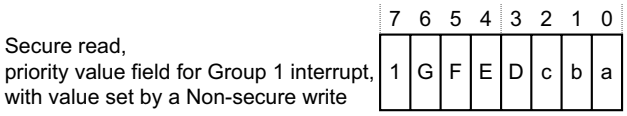


Figure 3-5 Secure read of the priority field for a Group 1 interrupt

A Sec e e e a e f e d f a G 1 e ca e b [7] 0, b ee *R* a e 3-56. If a Sec e e e b [7] 0:

A N - ec e ead e e a e `0bGFEDcba0`.

A N - ec e e ca c a e e a e f e f e d, b a a e a a b [7] e 1 e D b e f e f e d.

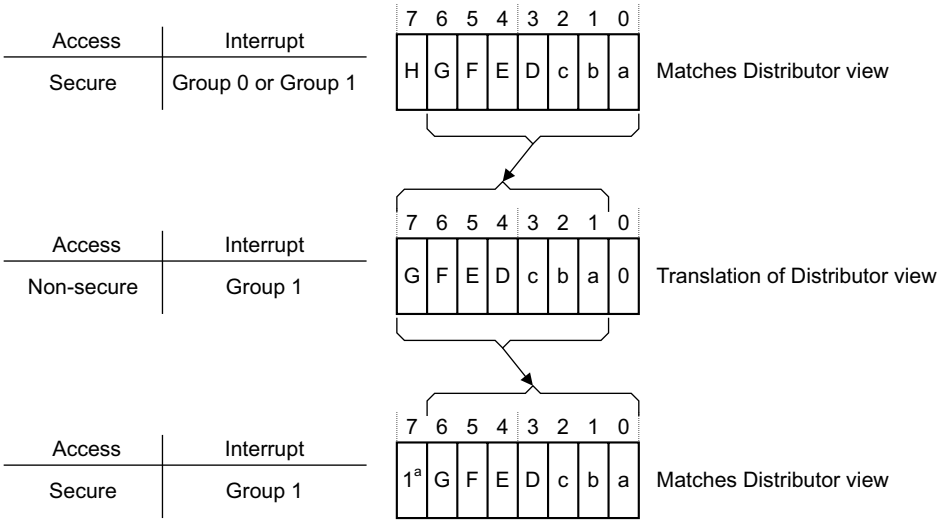
Note

T be a f N - ec e acce e a e e P a e f e d e GICD_IPRIORITYR :

f e P f e d e GICC_PMR d a a e b [7] == 0, e e f e d RAZ/WI N - ec e acce e

f e P f e d e GICC_RPR d a a e b [7] == 0, e e f e d RAZ N - ec e ead .

F e 3-6 a e 3-55 e e a be ee e e f e P a e f e d .

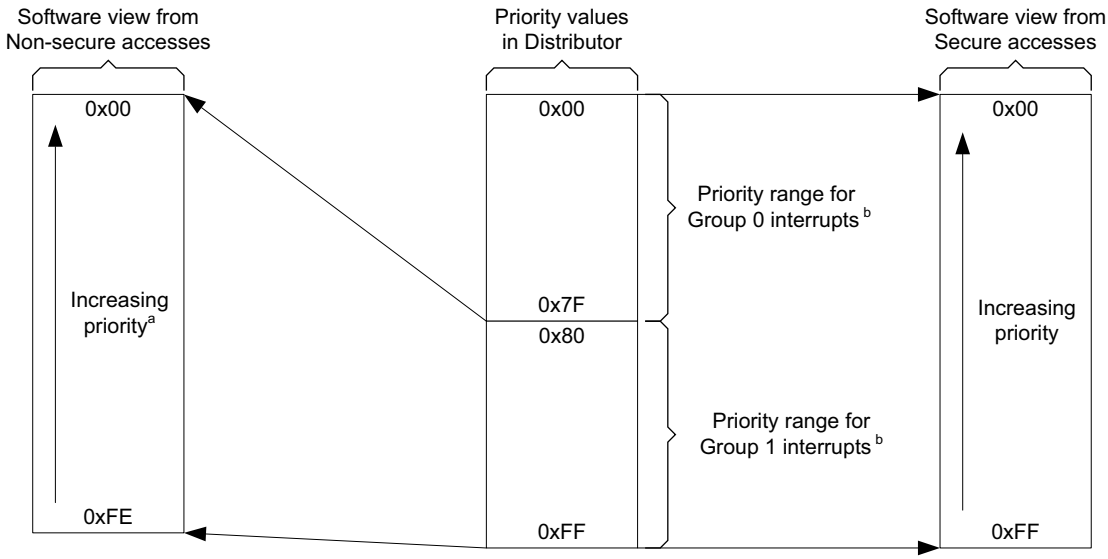


a If the priority value was set by a Non-secure write, bit [7] is set to 1 in the Distributor, and a Secure read sees this value. A Secure write to the field can set this bit to 0, see text for how this affects Non-secure accesses to the field.

The priority field for a Group 0 interrupt is RAZ/WI to Non-secure accesses.

Figure 3-6 Relationship between Secure and Non-secure views of interrupt priority fields

F e 3-7 e f a e e f e e e, f Sec e a d N - ec e acce e, e a e e a e e d eD b , a d e e a e a a e be Sec e a d N - ec e acce e. T f a GIC a e e e a a e f a e.



a All priority values are even (bit [0] == 0) in the view from Non-secure accesses.

b Ranges recommended by ARM. See text for more information, including about cases where these ranges might not be appropriate.

Figure 3-7 Software views of the priorities of Group 1 and Group 0 interrupts

Tab e 3-6 e be f a e b e e ed b e GIC affec e Sec e a d N - ec e e f e f a G 1 e .

Note

N - ec e f a e a b f e e f G 0 e .

Table 3-6 Effect of not implementing some priority field bits, with GIC Security Extensions

Implemented priority bits, as seen in Secure view	Possible priority field values, for a Group 1 interrupt	
	Secure view	Non-secure view
[7:0]	0xFF-0x00 (255-0), a a e	0xFE-0x00 (254-0), e e a e
[7:1]	0xFE-0x00 (254-0), e e a e	0xFC-0x00 (252-0), e f 4
[7:2]	0xFC-0x00 (252-0), e f 4	0xF8-0x00 (248-0), e f 8
[7:3]	0xF8-0x00 (248-0), e f 8	0xF0-0x00 (240-0), e f 16

T de f e e e a f a e e e f a e e e a a e e a f GIC a c ec e f c a e ded e a d e f e e e GIC e e e GIC Sec E e . H e e, a e e e a f a e a a a e e e e G 0 a d G 1 e . See *P GIC S E* a e 3-60 f e f a .

F e f a ab - e a ed e e acce e c a c a ed e GIC Sec E e , ee e e d c de *I GIC* a e 3-58.

Recommendations for managing priority values

ARM ec e d a :

f a G 0 e , f a e e b [7] f e a e f e d 0

f a Sec e e e e f a G 1 e , f a e e b [7] f e a e f e d 1.

T e e a a G 0 e a e e a e , a d e e f e e e , a a G 1 e . H e e , a e a e e e e a ca be e c e e, ee *P GIC S E* a e 3-60.

Note

W e b e GIC a d e c ec e d ce c de e Sec E e , G 0 e a e Sec e e , a d G 1 e a e N - ec e e e .

S f a e a e a a e e f e GIC Sec E e , a d e e f e e e a Sec e N - ec e acce e GIC e e . H e e, f a e e e d e , f a e ca e 0xFF ec e d GICD_IPRIORITYR a e f e d, a d e ead bac e a e e d e f e d de e e e e d e a e. ARM ec e d a , bef e c ec e a e a :

G f a e e a e , f a e f d ab e e e

G f a SGI, f a e f c ec a e e ac e.

3.5.2 Control of preemption by Group 1 interrupts

See *P* a e 3-45 a d *P* a e 3-45 f e f a ab ee .

W e a GIC e e a e , e GICC_BPR a a ed de e e e a G 0 e a ed e ce , f b e ee . B defa , e GICC_ABPR ed de e e e a G 1 e a ed f b e ee . H e e, e GICC_CTLR.CBPR e 1, GICC_BPR ed f de e b e ee , f b G 0 a d G 1 e .

Effect of the GIC Security Extensions on control of preemption by Group 1 interrupts

If e GIC e e a c de e Sec E e :

e CBPR b e e ed e Sec e c f GICC_CTLR.

e Sec e c f GICC_BPR a :

G a a ed de e e e e G 0 e a e a ed e ce

G e GICC_CTLR.CBPR e 1, a ed de e e e e G 0 e a e a ed GICC_ABPR a a a f e N - ec e c f GICC_CTLR

GICC_ABPR a Sec e e e , acce b e b Sec e f a e acce e .

3.5.3 The effect of interrupt grouping on priority grouping

W e a e e GICC_ABPR, e effec e b a a e e e a a ed e e e , a Tab e 3-7 . T ea a f a e a a e e f e effec f e a d e GIC Sec E e ee e a e ec a e a d e f e e a ce a Sec e N - ec e a e.

Note

I GIC 2, e effec e b a a e ad e a cc GIC e e a a d c de e Sec E e .

P a a e a e e a e e d e D b , e a e b e a N - ec e ead f e a e c e d a N - ec e e . See F e 3-6 a e 3-55 a d F e 3-7 a e 3-55.

T e b a a e ed f e GICC_ABPR e e :

IMPLEMENTATION DEFINED

e a e 1-4

e ea e a e a e ed f e Sec e c f e GICC_BPR e e .

Tab e 3-7 e e a f G 1 e e GICC_CTLR.CBPR==0.

Table 3-7 Priority grouping for Group 1 interrupts when GICC_CTLR.CBPR==0

GICC_ABPR value	Interrupt priority field [7:0]		
	Group priority field	Subpriority field	Field with binary point[a]
0[b]	-	-	-.
1	[7:1][c]	[0]	HGFEDcb.
2	[7:2][c]	[1:0]	HGFEDc.
3	[7:3][c]	[2:0]	HGFED.
4	[7:4][c]	[3:0]	HGFE.
5	[7:5][c]	[4:0]	HGF.

Table 3-7 Priority grouping for Group 1 interrupts when GICC_CTLR.CBPR==0 (continued)

GICC_ABPR value	Interrupt priority field [7:0]		
	Group priority field	Subpriority field	Field with binary point[a]
6	[7:6][c]	[5:0]	HG.
7	[7][c]	[6:0]	H.

a. G abe a a F e 3-6 a e 3-55.
b. N ed.
c. If a N - ec e e e a e f e d f a N - ec e e e b [7] 1.

F G 0 e , e be a a de c bed *P* a e 3-45.

I a GIC e e a a c de e Sec E e , e GICC_CTLR.CBPR == 1:

A N - ec e ead f e GICC_BPR e e a e f e Sec e GICC_BPR, c e e ed b 1, a d a a ed `0b111`.

N - ec e e GICC_BPR a e ed

e GICC_ABPR e e ed da .

3.5.4 Interrupt generation when the GIC supports interrupt grouping

T e e d c de *E* , a e 3-64 de c be e e e a f e b e GIC e e GIC e .

3.6 Additional features of the GIC Security Extensions

T a e 3-48 a d *I* a e 3-53 de c be a fea e f e GIC Sec E e , e ec a f a GIC 1 e e a , e e e ed a a f e GIC Sec E e . T ec de c be e e fea e f e GIC Sec E e .

S f a e ca de ec f e GIC Sec E e b ead e GICD_TYPER.Sec E b , ee *I C T R , GICD T PER* a e 4-88.

Note

I e c e f a GIC a e e e GIC Sec E e c ec ed a ce a e e e ARM Sec E e , G 0 e a e Sec e e , a d G 1 e a e N - ec e e . See *S E* a e 1-16 f e f a .

I add :

T e ba f e e de de e de c f Sec e a d N - ec e e , ee *E GIC S E* a e 4-80.

T e N - ec e c f e GICC_BPR a a ed a e GICC_ABPR. T a Sec e e e , ea acce b e b Sec e acce e .

3.6.1 Access from processors not implementing the ARM Security Extensions

W e c ec a ce a d e e ARM Sec E e a GIC a e e e GIC Sec E e , ca a ce acce e e GIC a e a ed a e e Sec e N - ec e:

F a ce a Sec e acce e :

G T e ce ca c a a ec f e GIC, a d e ef e ca a e c f a c a e a affec Sec e f a e e ce .

G I a GIC 2 e e a , e ce e Sec e acce e a a ed e e , c a e GICC_AIAR, ce G 1 e .

G Beca e GIC 1 e e a d c de e a a ed e e , f e e e a e e e ce a e e de eca ed GICC_CTLR.Ac C b e ab e G 1 e be ce ed e a da d CPU e face e e .

F a ce a N - ec e acce e :

G T e ce ca c G 0 e . F e GIC be a ed, e e e e a c de a ea e ce a ca a e Sec e acce e .
A e e a Sec e ce e f Sec e acce e be a f f a N - ec e ce . T a e de b e f e GIC e e de a e d f e Sec e ce acce ce -ba ed c e f e e a be e N - ec e ce .

G T e a N - ec e ce c G 0 e , a GIC 2 e e a ca e e e GICD_NSACR e e . A e e a f e e e e a Sec e ce e e e f N - ec e acce e f a a c a ce c e a ec f e e a f e G 0 SGI a d SPI .

G A GIC e e a ca c f e e GICD_IGROUPR e e a e a e a e G 1 e e e. ee *GICD IGRO PR0* a e 4-92 f e f a .

3.6.2 The effect of the GIC Security Extensions on priority masking

T ec de c be e GIC Sec E e c a e e f a e *P* a e 3-45.

If e GIC e e e GIC Sec E e , e GICC_PMR RAZ/WI N - ec e acce e f d a a e b [7] == 0. I a e a , N - ec e f a e d e acce e GICC_PMR e a ed c a a e. F e f a ee *N - G 0* a e 4-81.

3.6.3 Priority management and the GIC Security Extensions

A e a e e e GIC Sec E e ca e e f c e e f a a e :

Non-cooperative A Sec e e a e e a a N - ec e e , a d ca a a ee a N - ec e e .

Co-operative Sec e a d N - ec e f a e e ac a e Sec e e e a e N - ec e e .

Sec e f a e f a e e ec a ce a e e e ARM Sec E e , a ca a e Sec e acce e e GIC, a d be ab e a e N - ec e acce e . N - ec e f a e ca a e N - ec e acce e .

W e e Sec e f a e a a e e e e fa N - ec e e , a e e b [7] f e a e fe d e 1, a e f e e e e af f e e e ed a e. H e e, a e a e e e fa N - ec e e a a e e e af f e e e ed a e, f e a e a a e a SGI f N - ec e f a e a a e a ce a e ec e Sec e f a e.

Sec e f a e ca a e e fa Sec e e a a e e e af f e e e ed a e, a a e a e N - ec e e .

Note

Se e fa Sec e e e e af f e a e de a f ec a ac , c a de a f e ce. Sec e f a e c de e b fa ac f d bef e e a Sec e e a a e e a e be N - ec e f a e.

T e GIC a c ec ed e e e a ce e e e e a e c e e f a a e .

3.7 Pseudocode details of interrupt handling and prioritization

T e f ec de e d c de de c f e a d a d a , a d e GIC Sec E e , a d de c be e acce e e e a c a a e a e e e GIC Sec E e :

G
E a e 3-64
T *GIC S* *E* a e 3-66.

3.7.1 General helper functions and definitions

T e f e d c de de e e f c a d def ed e e e e GIC e d c de:

```
// Helper functions
// ================

SignalFIQ(boolean next_fiq, integer cpu_id) // Signals an interrupt on the FIQ input to the
                                            // processor, according to the value of next_fiq.

SignalIRQ(boolean next_irq, integer cpu_id) // Signals an interrupt on the IRQ input to the
                                            // processor, according to the value of next_irq.

boolean IsGrp0Int(integer InterruptID, cpu_id)
                                            // Returns TRUE if the field in the GICD_IGROUPRn
                                            // register associated with the argument InterruptID
                                            // is set to 0, indicating that the interrupt is
                                            // configured as a Group 0 interrupt.

boolean IsEnabled(integer InterruptID, cpu_id)
                                            // Returns TRUE if the interrupt specified by the
                                            // argument InterruptID is enabled in the associated
                                            // GICD_ISENABLERn or GICD_ICENABLERn register.

bits(3) SGI_CpuID(integer InterruptID, cpu_id)
                                            // Returns the ID of a source CPU for a pending interrupt
                                            // with the given interruptID targeting the current
                                            // CPU. If there are multiple source CPUs, the one
                                            // chosen is IMPLEMENTATION SPECIFIC.

bits(8) ReadGICD_ITARGETSR(integer InterruptID, cpu_id)
                                            // Returns an 8-bit field specifying which CPUs should
                                            // receive the interrupt specified by argument InterruptID

boolean AnyActiveInterrupts(integer cpu_id) // Returns TRUE if any interrupts are active on this
                                            // processor.

bits(8) ReadGICD_IPRIORITYR(integer InterruptID, cpu_id)
                                            // Returns the 8-bit priority field from the
                                            // GICD_IPRIORITYR associated with the argument InterruptID.

WriteGICD_IPRIORITYR(integer InterruptID, cpu_id, bits(8) Value)
                                            // Updates the priority field in the GICD_IPRIORITYR
                                            // associated with the argument InterruptID with the 8-bit
                                            // Value.

IgnoreWriteRequest()                        // Ignore the register write request (no operation).

AcknowledgeInterrupt(integer InterruptID, cpu_id)
                                            // Set the active state and attempt to clear the pending
                                            // state for the interrupt associated with the argument
                                            // InterruptID

// Global variables
// ================

integer cpu_id                  // An identifier for a specific CPU Interface. The value of this
```

```
                                // variable has implicit effects on which CPU interface register,
                                // CPU interface signal or banked version of a Distributor
                                // register is accessed.

boolean NS_access               // current GIC access state:
                                //   TRUE:  Non-secure
                                //   FALSE: Secure.

// NOTE: Architected registers are considered global variables identified
//       by their architecture mnemonic, and as such are not declared here.


// global constants
// ================

integer    MINIMUM_BINARY_POINT   // A minimum binary point value of 0,1,2 or 3,
                                  // this is an IMPLEMENTATION DEFINED value.
                                  // NOTE: min. value is the SECURE value where supported

boolean    IGNORE_GROUP_ENABLE    // IMPLEMENTATION DEFINED boolean that determines whether the
                                  // highest priority pending interrupt is masked by the distributor
                                  // enable BEFORE or AFTER prioritisation:
                                  //
                                  //   BEFORE prioritisation        Value = FALSE
                                  //   AFTER prioritisation         Value = TRUE

boolean    GICC_MASK_HPPIR        // IMPLEMENTATION DEFINED boolean that determines whether a read
                                  // of GICC_HPPIR returns a spurious interrupt for pending
                                  // interrupts disabled by GICC_CTLR.EnableGrp{0,1}} == '0'

bits(8)    P_MASK                 // IMPLEMENTATION DEFINED mask of valid priority bits:
                                  //   Consists of an 8-bit field where the top N bits are set to 1,
                                  //   where N is the number of priority bits implemented.
                                  //   For systems without the Security Extensions, supported
                                  //   values are 0xF0, 0xF8, 0xFC, 0xFE and 0xFF.
                                  //   For systems with the Security Extensions, supported
                                  //   values are 0xF8, 0xFC, 0xFE and 0xFF.

// PriorityIsHigher()
// ==================

boolean PriorityIsHigher(bits(8) pr1, bits(8) pr2)
    return UInt(pr1) < UInt(pr2);        // Lower number represents higher priority.

// GIC_PriorityMask()
// ==================

// NOTE: where the Security Extensions are not supported, NS_mask = '0'

bits(8) GIC_PriorityMask(integer n, bit NS_mask) // Calculate the Binary Point (group) mask.
    assert n >= 0 && n <= 7;                     // Range check for the priority mask.

    if NS_mask == '1' then                       // Mask generation for a secure GIC access.
        n = n - 1;
                                            // CHECK:
    if n < MINIMUM_BINARY_POINT then        // Saturate n on the minimum value supported; range 0-3
        n = MINIMUM_BINARY_POINT;           // NOTE: min. value is the SECURE value where supported

    mask = '1111111100000000'<14-n:7-n>;    // Generate the 8-bit group priority mask.
    return mask;

// boolean IsPending()
// ===================
//
// Returns TRUE if the interrupt specified by argument interruptID
// is pending for the CPU specified by argument cpuID
//
```

```
boolean IsPending(integer interruptID, integer cpuID)
    pending = FALSE;

    target_cpus = ReadGICD_ITARGETSR(interruptID);

    if PEND && !ACTIVE(interruptID) && target_cpus<cpuID> == '1' then
        pending = TRUE;

    return pending;

// HighestPriorityPendingInterrupt()
// =================================
//
// Returns the ID of the highest priority interrupt that is pending and enabled.
// Otherwise, returns 1023 (i.e. a spurious interrupt)
//
// In implementations where interrupts are masked by the distributor group enable bits AFTER
// prioritisation (i.e. IGNORE_GROUP_ENABLE is TRUE), this function may return the ID of a pending
// interrupt in a disabled group even though there is a (lower priority) pending interrupt that is
// fully enabled (i.e. enabled in GICD_IENABLER and the appropriate group enable bit is '1' in
// GICD_CTLR). This is a helper function only and does not explain the full efect of GICC_HPPIR.
// The value returned by a read of GICC_HPPIR is explained in the pseudocode provided with the
// register description.

bits(10) HighestPriorityPendingInterrupt(integer cpu_id)

    num_interrupts = 32 * (UInt(GICD_TYPER<4:0>) + 1); // Work out how many interrupts are supported

    hppi= 1023;                                        // Set initial ID to be no intterupt pending

    for intID = 0 to num_interrupts - 1
        group_enabled = ( IsGrp0Int(intID) && (GICD_CTLR.EnableGrp0 == '1')) ||
                        (!IsGrp0Int(intID) && (GICD_CTLR.EnableGrp1 == '1'));

        if IsPending(intID, cpu_id) && IsEnabled(intID) then
            if group_enabled || IGNORE_GROUP_ENABLE then
                if PriorityIsHigher(ReadGICD_IPRIORITYR(intID), ReadGICD_IPRIORITYR(hppi)) then
                    hppi = intID;

    return(hppi);
```

3.7.2 Exception generation pseudocode

I e , a d e GIC Sec E e , a e e ce e e a de f ca e c ca ed:

E , de c be e ce e e a b a GIC e e a a e , a d c de e Sec E e

E , a e 3-65 de c be e f ed e ce e e a de f a GIC e e a a d e e .

Exception generation pseudocode, with interrupt grouping

T e f e d c de de c be e ce a e e e a ed b a CPU e face a e e e GIC Sec E e :

```
// GenerateExceptions()
// ====================
//

GIC_GenerateExceptions(
    boolean systemFIQ,
    boolean systemIRQ)

    while TRUE do                               // Loop continuously.
        cpu_count = UInt(GICD_TYPER<7:5>) + 1; // Determine the number of CPU interfaces.

        for cpu_id = 0 to cpu_count - 1         // Loop though CPU interfaces. The iterations of
                                                // this loop are permitted to occur in parallel.

            (next_int, next_grp0) = UpdateExceptionState(cpu_id);  // Returns pending interrupts, masked
                                                // by distributor enables but not cpu i/f enables

            irq_wake = FALSE;                   // IRQ wake up signal to power management, if required
            fiq_wake = FALSE;                   // FIQ wake up signal to power management, if required

            cpu_irq = FALSE;                    // IRQ signal to CPU
            cpu_fiq = FALSE;                    // FIQ signal to CPU

            if (next_int) then
                if (next_grp0 && GICC_CTLR[cpu_id].FIQEn == '1') then
                    fiq_wake = TRUE;
                    if (GICC_CTLR[cpu_id].EnableGrp0 == '1') then
                        cpu_fiq = TRUE;
                else
                    irq_wake = TRUE;
                    if ( next_grp0 && GICC_CTLR[cpu_id].EnableGrp0 == '1' ||
                        !next_grp0 && GICC_CTLR[cpu_id].EnableGrp1 == '1')
                    then
                        cpu_irq = TRUE;

// Optional bypass logic
//
            if GICC_CTLR[cpu_id].EnableGrp0 == '0' || GICC_CTLR[cpu_id].FIQEn == '0'
            then
                if GICC_CTLR[cpu_id].FIQBypDisGrp0 == '0' ||
                   (GICC_CTLR[cpu_id].FIQBypDisGrp1 == '0' && GICC_CTLR[cpu_id].FIQEn == '0')
                then
                    cpu_fiq = systemFIQ;                        // Set FIQ to bypass

            if GICC_CTLR[cpu_id].EnableGrp1 == '0' &&
               (GICC_CTLR[cpu_id].EnableGrp0 == '0' || GICC_CTLR[cpu_id].FIQEn == '1')
            then
                if GICC_CTLR[cpu_id].IRQBypDisGrp1 == '0' ||
                   (GICC_CTLR[cpu_id].IRQBypDisGrp0 == '0' && GICC_CTLR[cpu_id].FIQEn == '1')
                then
                    cpu_irq = systemIRQ;                        // Set IRQ to bypass
```

```
//
// End, optional bypass logic

            SignalFIQ(cpu_fiq, cpu_id);                         // Update driven status of FIQ.
            SignalIRQ(cpu_irq, cpu_id);                         // Update driven status of IRQ.


// UpdateExceptionState()
// ======================
//

(boolean, boolean) UpdateExceptionState(integer cpu_id)

    sbp = UInt(GICC_BPR[cpu_id]<2:0>);                    // Secure version of this register.
    nsbp = UInt(GICC_ABPR[cpu_id]<2:0>);

    next_int = FALSE;
    next_grp0 = FALSE;

    intID = HighestPriorityPendingInterrupt(cpu_id);       // Establish the ID of the highest pending
                                                           // interrupt on the this CPU interface.

    if PriorityIsHigher(ReadGICD_IPRIORITYR(intID), GICC_PMR[cpu_id]<7:0>) &&
       IsPending(intID, cpu_id)
    then
        smsk = GIC_PriorityMask(sbp, '0');
        if GICC_CTLR[cpu_id].CBPR == '1' then
            nsmsk = smsk;
        else
            nsmsk = GIC_PriorityMask(nsbp, '1');

        if IsGrp0Int(intID) &&                            // Highest pending interrupt is secure
           (GICD_CTLR.EnableGrp0 == '1')                    // and secure interrupts are enabled
        then
            if !AnyActiveInterrupts() ||
               PriorityIsHigher(ReadGICD_IPRIORITYR(intID), GICC_RPR[cpu_id]<7:0> AND smsk)
            then
                next_int = TRUE;
                next_grp0 = TRUE;
        else
            if (!IsGrp0Int(intID)) &&                     // Highest pending interrupt is non-secure
               (GICD_CTLR.EnableGrp1 == '1')                // and non-secure interrupts are enabled
            then
                if !AnyActiveInterrupts() ||
                   PriorityIsHigher(ReadGICD_IPRIORITYR(intID), GICC_RPR[cpu_id]<7:0> AND nsmsk)
                then
                    next_int = TRUE;
                    next_grp0 = FALSE;

    return(next_int, next_grp0);
```

Exception generation pseudocode, when interrupt grouping is not supported

T e f e d c de de c be e ce a e e e a ed b a GIC a d e e
. T ea a e a GIC 1 e e a a d e c de e Sec E e .

```
// GenerateExceptions()
// ====================
//

GIC_GenerateExceptions()
    while TRUE do                                  // Loop continuously.
        cpu_count = UInt(GICD_TYPER<7:5>) + 1;     // Determine the number of CPU interfaces.

        for cpu_id = 0 to cpu_count - 1            // Loop though CPU interfaces. The iterations of
                                                   // this loop are permitted to occur in parallel.
            next_irq = UpdateExceptionState(cpu_id);
```

```
            SignalIRQ(next_irq, cpu_id);            // Update driven status of IRQ.


// UpdateExceptionState()
// ======================
//

boolean UpdateExceptionState(integer cpu_id)

    next_irq = FALSE;

    intID = HighestPriorityPendingInterrupt(cpu_id);  // Establish the ID of the highest pending
                                                      // interrupt on the this CPU interface.

    if PriorityIsHigher(ReadGICD_IPRIORITYR(intID), GICC_PMR[cpu_id]<7:0>) &&
       IsPending(intID, cpu_id)
    then
        if GICD_CTLR.Enable == '1' && GICC_CTLR.Enable == '1' then
            mask = GIC_PriorityMask(GICC_BPR[cpu_id]<2:0>, '0');

            if !AnyActiveInterrupts() ||
               PriorityIsHigher(ReadGICD_IPRIORITYR(intID), GICC_RPR[cpu_id]<7:0> AND mask)
            then
                next_irq = TRUE;

    return(next_irq);
```

3.7.3 The effect of the GIC Security Extensions on accesses to prioritization registers

T e GIC Sec E e c a e e f e be a f acce e e a e e . See e e d c de f c :

I P R , GICD IPRIORIT R a e 4-104
I P M R , GICC PMR a e 4-131
B P R , GICC BPR a e 4-133
I A R , GICC IAR a e 4-135
R P R , GICC RPR a e 4-142
H P P I R , GICC HPPIR a e 4-143.

See *N - G 0* a e 4-81 f e f a .

3.8 The effect of the Virtualization Extensions on interrupt handling

I e ea, eV a a E e a e effec e GIC a d e a d e ca e .
See C a e 5 *GIC S* f f a ab e GIC V a a E e
a e a d .

3.9 Example GIC usage models

A ARM ce a e e e ARM 7-A ARM 7-R a c ec e f e e e e a , **nIRQ** a d **nFIQ**, eac a a c a ed e ce a d ce de:

A e a IRQ e e e e a e a IRQ e ce . B defa , a e IRQ de, a d a e e ce a b e e IRQ e ce .

A e a FIQ e e e e a e a FIQ e ce . B defa , a e FIQ de, a d a e e ce a b FIQ a d IRQ e ce .

T e f ec de c be d ffe e GIC a e de , a ee ec f c e e e e :
IRQ FIQ N - S
S IRQ FIQ S E a e 3-70.
S IRQ FIQ a e 3-71.

A f e e a e de e a e e e ad a e e e a F e 3-8, a GIC a G 0 a d G 1 e .

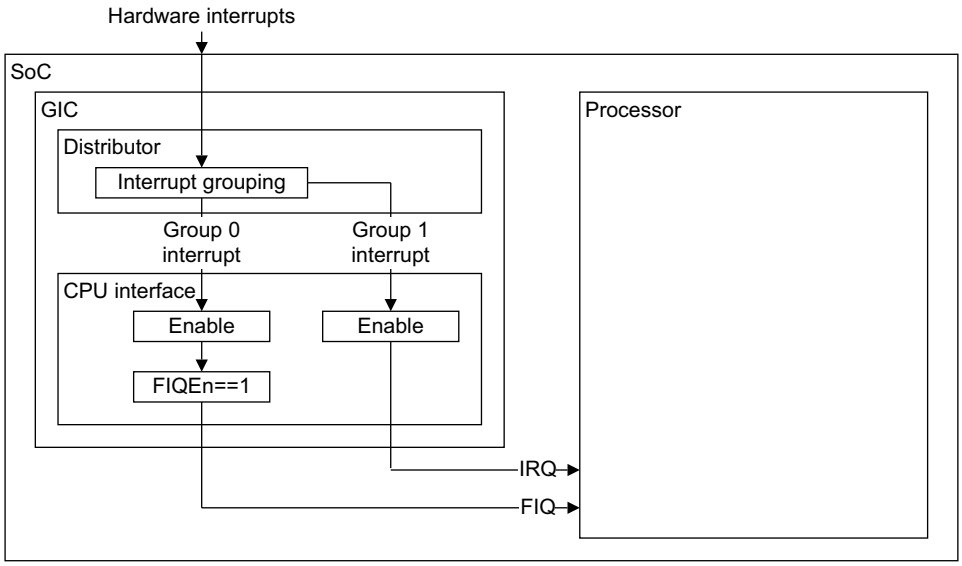


Figure 3-8 Generic GIC usage model

I eac a e de , f a e e e GICD_IGROUPR e e a e e , a ed e ce e IRQ a d FIQ e e e .

Note

T e a e de de c bed *S IRQ FIQ* a e 3-71 a e e e GIC e e e GIC V a a E e .

3.9.1 Using IRQs and FIQs to provide Non-secure and Secure interrupts

F e 3-9 a e 3-69 a e a e e e GIC Sec E e , c ec ed a ce a e e e ARM ce Sec E e . T e e a :
e G 0 e a Sec e e , a ed a FIQ
e G 1 e a N - ec e e , a ed a IRQ .

T ea a , e ce , FIQ e a e e e ed N - ec e f a e, a d IRQ e a e e e ed Sec e f a e.

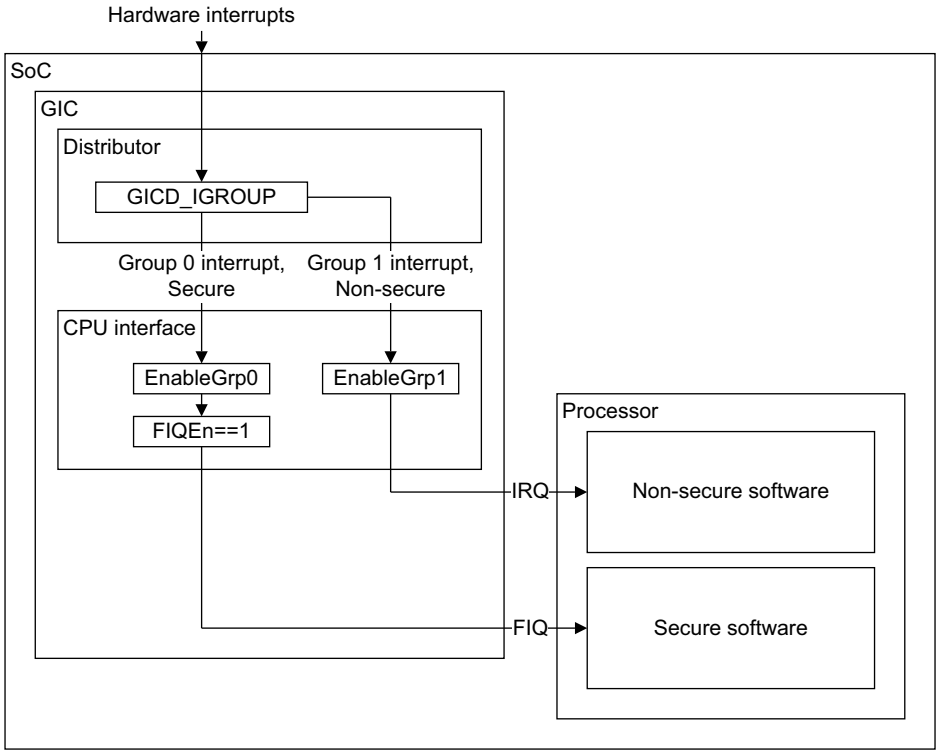


Figure 3-9 Using the GIC to route Secure and Non-secure interrupts

Note

T e e f G 0 a d G 1 e a Sec e e a FIQ , a d N - ec e e a IRQ , e e e ce :
e FIQ be a e Sec e M de
e e N - ec e f a e f a FIQ
e e a IRQ a e a ed e e e e a Sec e a e.

Controlling Secure and Non-secure interrupts independently

T e e F e 3-9 f f e e e a ec e e e a N - ec e e a e fe e Sec e e a . Sec e f a e a e f c f FIQ b e e Sec e f a e a d e e N - ec e f a e a e .

O a GIC e e , a e a e a ed G 0, a e Sec e e . Sec e f a e e ce :

a e GICD_IGROUPR e e d ca e c e a e G 1, N - ec e e e Sec e GICC_CTLR.FIQE b 1 c f e e CPU e face e FIQ f G 0 e .

e ab e G 0 e a d G 1 e , de e de , e D b :

- G GICD_CTLR.E ab e G 0 e ab e G 0 e
- G GICD_CTLR.E ab e G 1 e ab e G 1 e .

e ab e G 0 e a d G 1 e , de e de , e CPU e face:

- G GICC_CTLR.E ab e G 0 e ab e G 0 e
- G GICC_CTLR.E ab e G 1 e ab e G 1 e .

3.9.2 Supporting IRQs and FIQs when not using the processor Security Extensions

F e 3-10 a e c e ce d e e e , , e P ce Sec E e . T e ca e e e ded b e GICD_IGROUPR e e c b IRQ a d FIQ , ba ed :

a FIQ e G 0
a IRQ e G 1.

T ec a e a GIC 1 e e a a c de e GIC Sec E e , a GIC 2 e e a .

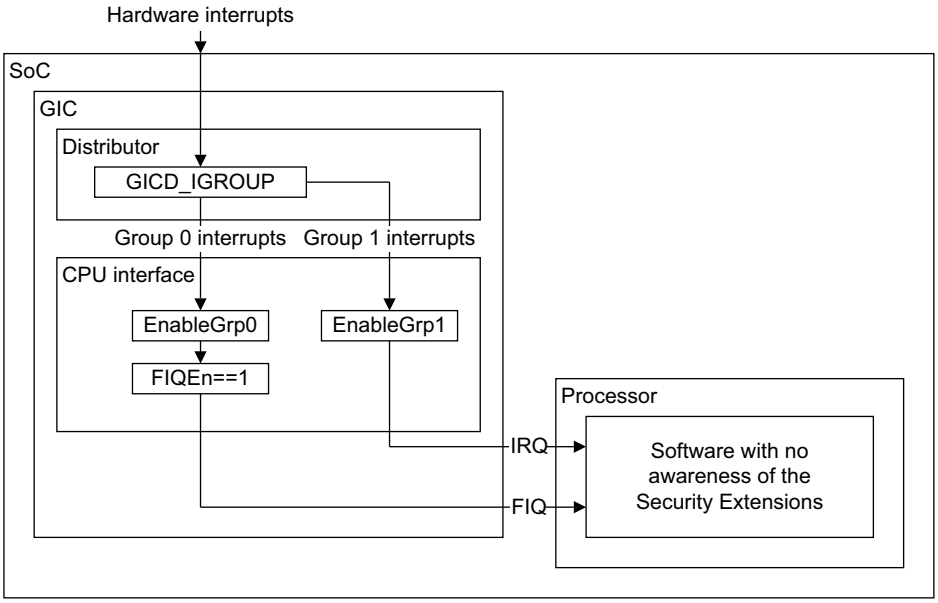


Figure 3-10 Using interrupt grouping to route IRQs and FIQs

O a GIC e e , f a GIC e e a a e , a e a ed G 0. T e ef e, e c f a , f a e e ec e ce :

P a e GICD_IGROUPR e e a IRQ e G 1.

Note

F GIC 2 e e a a d c de e Sec E e , e GICD_IGROUPR e e a e a e IMPLEMENTATION DEFINED, ee *I G R , GICD IGRO PR* a e 4-91.

Se GICC_CTLR.FIQE 1, a G 0 e FIQ.

Se GICC_CTLR.Ac C 0, a b FIQ a d IRQ e a e ac ed ed f e e add e ace, :

G e GICC_IAR ac ed e a G 0 e
G e GICC_AIAR ac ed e a G 1 e
G e GICC_EOIR d ca e c e f a G 0 e
G e GICC_AEOIR d ca e c e f a G 1 e .

H e e , GICC_AIAR a d GICC_AEOIR a e e e ed a GIC 2 e e a . A ce e a a GIC 1 e e a a e e e de eca ed de f e a GICC_CTLR.Ac C e 1.

C f e e e ed b a de , b e e :

G e GICC_CTLR.CBPR 0, a G 0 e GICC_BPR, a d G 1 e GICC_ABPR

G e GICC_CTLR.CBPR 1, a G 0 a d G 1 e a c b a e e, GICC_BPR.

3.9.3 Supporting IRQs and FIQs in a virtualized processor environment

F e 3-11 a e 3-72 a e a ce a a , e e ec f e ac f a e a ac e . T e ba f e ce a e de :

Sec e f a e a :

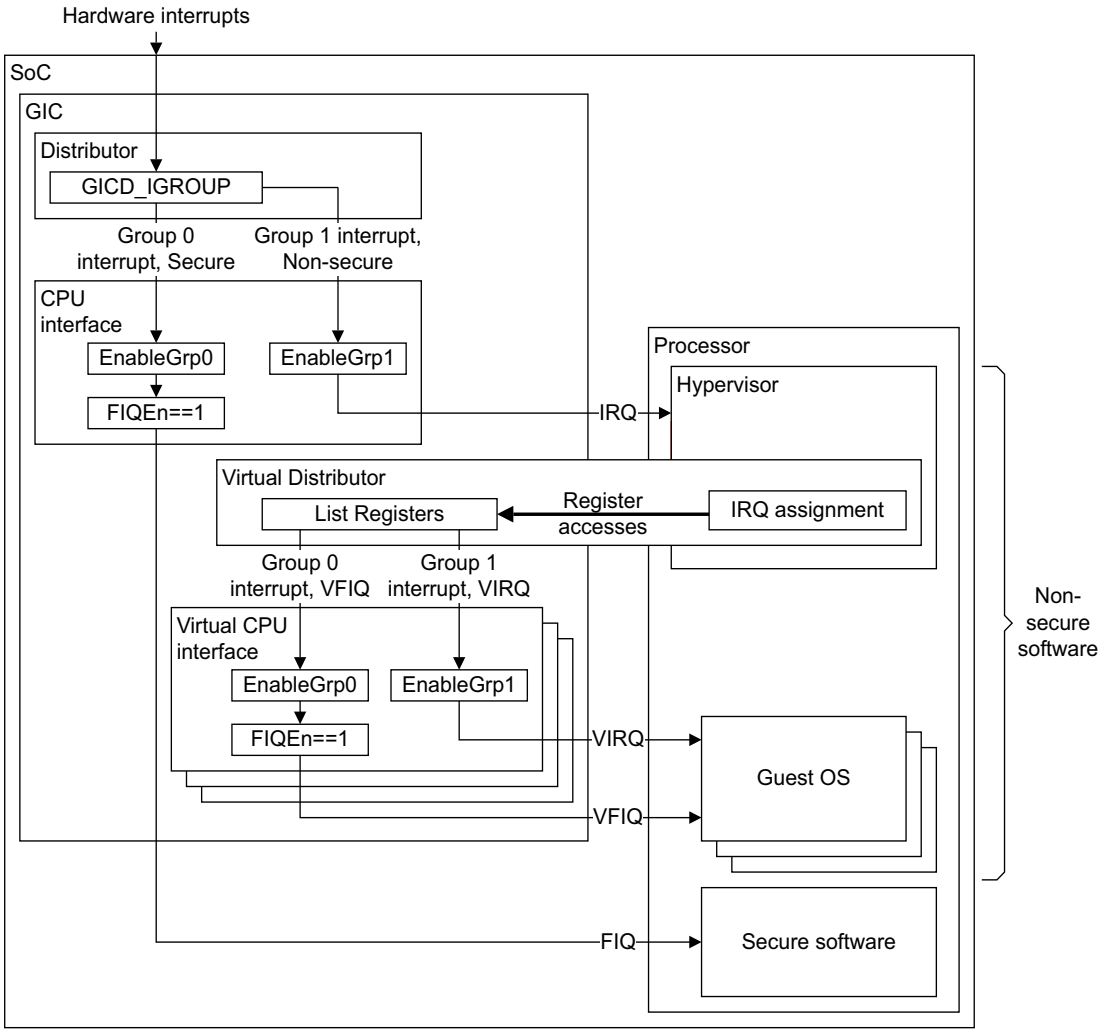
G Sec e e G 0, a ed e ce a FIQ

G N - ec e e G 1, a ed e ce a IRQ .

T e a e de de c bed *IRQ FIQ N - S* a e 3-68.

A e :

G I e e a , fea e f e V a



VFIQ is Virtual FIQ, VIRQ is Virtual IRQ

Figure 3-11 Using the GIC in a virtualized system

W e e GIC a a IRQ e ce , e e ed H de. T e e de e e e e e f e f, f a G e OS. If f a G e OS de e e :

c G e OS a d e e e

e e a G e OS a c f ed e e a a FIQ a a IRQ

e e , ba ed e c f a b e a e G e OS.

If e e a e ec e G e OS, e e da e eL e e , add e e e f e d e f ec e a ac e.

Note

O ece a IRQ a ca be a d ed b ec e G e OS, e e ca e e :

G a fe c a G e OS a ca a d e e e

G a e e a e d , a a f e a ed c e f e a a e G e OS.

A e ca a e e e a ca be a d ed b e a e G e OS, a d e e a be ed a ec f c G e OS.

A G e OS a d e a a e e ac a d a d e ec e d ca e . T e G e OS ca de ec a a d a a e a e a a ca e .

Programmers' Model

T c a e de c be e D b a d CPU e face e e . I c a e f ec :

4.1 About the programmers' model

T e a e de de e f a e e face e GIC. T c a e de c be e a e de f e GIC D b a d CPU e face , a e a e a e - a ed e e e face.

T e f ec de c be e a e de :

GIC
D
CP a e 4-76
GIC a e 4-77
E *D* *CP* a e 4-77
E *GIC S* *E* a e 4-80.

Tab e 4-1 a e 4-75 a d Tab e 4-2 a e 4-76 de c be e e e acce e a f :

RW Read a d e.
RO Read . W e a e ed.
WO W e . Read e a UNKNOWN a e.

Note

T ec d e de c be e a e de f e GIC a e face c e e a d e a CPU e face , a e GIC V a a E e add a GIC e e a . See C a e 5 *GIC S* f e de c f e add e a e de a GIC a e e e GIC V a a E e .

4.1.1 GIC register names

A f e GIC e e a e a e a de a e c f e f c f e e e. I e e a e: e f ee e e a e GIC, d ca a GIC e e e f e e e f:

G D, d ca a D b e e
G C, d ca a CPU e face e e
G H, d ca a a e face c e e , ca acce ed b a e
G V, d ca a a CPU e face e e .

e e a e e a e a e c f e e e , f e a e e GIC D b C Re e ca ed GICD_CTLR.

Note

C a e 5 *GIC S* de c be e GICH_* a d GICV_* e e .

4.1.2 Distributor register map

Tab e 4-1 a e 4-75 e D b e e a . Add e ff e a e e a e e *D* def ed b e e e a . A GIC e e a e 32-b de. Re e ed e e add e e a e RAZ/WI.

Note

F e f a ab e ac e e a e , ee A e d B *R N* .

Table 4-1 Distributor register map

Offset	Name	Type	Reset[a]	Description
0x000	GICD_CTLR	RW	0x00000000	D b C Re e
0x004	GICD_TYPER	RO	IMPLEMENTATION DEFINED	I e C e T e Re e
0x008	GICD_IIDR	RO	IMPLEMENTATION DEFINED	D b I e e e Ide f ca Re e
0x00C-0x01C	-	-	-	Re e ed
0x020-0x03C	-	-	-	IMPLEMENTATION DEFINED e e
0x040-0x07C	-	-	-	Re e ed
0x080	GICD_IGROUPR [b]	RW	IMPLEMENTATION DEFINED[c]	I e G Re e
0x084-0x0FC			0x00000000	
0x100-0x17C	GICD_ISENABLER	RW	IMPLEMENTATION DEFINED	I e Se -E ab e Re e
0x180-0x1FC	GICD_ICENABLER	RW	IMPLEMENTATION DEFINED	I e C ea -E ab e Re e
0x200-0x27C	GICD_ISPENDR	RW	0x00000000	I e Se -Pe d Re e
0x280-0x2FC	GICD_ICPENDR	RW	0x00000000	I e C ea -Pe d Re e
0x300-0x37C	GICD_ISACTIVER [d]	RW	0x00000000	GIC 2 I e Se -Ac e Re e
0x380-0x3FC	GICD_ICACTIVER [e]	RW	0x00000000	I e C ea -Ac e Re e
0x400-0x7F8	GICD_IPRIORITYR	RW	0x00000000	I e P Re e
0x7FC	-	-	-	Re e ed
0x800-0x81C	GICD_ITARGETSR	RO[f]	IMPLEMENTATION DEFINED	I e P ce Ta e Re e
0x820-0xBF8		RW[f]	0x00000000	
0xBFC	-	-	-	Re e ed
0xC00-0xCFC	GICD_ICFGR	RW	IMPLEMENTATION DEFINED	I e C f a Re e
0xD00-0xDFC	-	-	-	IMPLEMENTATION DEFINED e e
0xE00-0xEFC	GICD_NSACR [e]	RW	0x00000000	N - ec e Acce C Re e , a
0xF00	GICD_SGIR	WO	-	S f a e Ge e a ed I e Re e
0xF04-0xF0C	-	-	-	Re e ed
0xF10-0xF1C	GICD_CPENDSGIR [e]	RW	0x00000000	SGI C ea -Pe d Re e
0xF20-0xF2C	GICD_SPENDSGIR [e]	RW	0x00000000	SGI Se -Pe d Re e
0xF30-0xFCC	-	-	-	Re e ed
0xFD0-0xFFC	-	RO	IMPLEMENTATION DEFINED	*I* a e 4-119

a. F de a f a e c a a e e e a e f IMPLEMENTATION DEFINED ca e ee e a a e e de c .
b. I a GIC 1 e e a , e e f e GIC e e e GIC Sec E e , e e RAZ/WI.
c. F e f a ee *GICD IGRO PR0* a e 4-92.
d. I GIC 1, e e a e e Ac e B Re e , ICDABR . T e e e e a e RO.

e. GIC 2 .
f. I a ce e e a , e e e e a e RAZ/WI.

4.1.3 CPU interface register map

Tab e 4-2 e CPU e face e e a . Add e ff e a e e a e e *CP* def ed b e e e a . A GIC e e a e 32-b de. Re e ed e e add e e a e RAZ/WI.

F a ce e e a , e GIC e e a e f CPU e face e e f eac CPU e face. ARM ec e d a eac ce a e a e CPU e face ba e add e f e CPU e face a c ec e GIC. T e a e CPU e face ba e add e f a ce . I IMPLEMENTATION DEFINED e e a ce ca acce e CPU e face e e f e ce e e .

Note

F e f a ab :
e e e added b e GIC V a a E e , ee C a e 5 *GIC S* e ac e e a e , ee A e d B *R N* .

Table 4-2 CPU interface register map

Offset	Name	Type	Reset	Description
0x0000	GICC_CTLR	RW	0x00000000	CPU I e face C Re e
0x0004	GICC_PMR	RW	0x00000000	I e P Ma Re e
0x0008	GICC_BPR	RW	0x0000000x[a]	B a P Re e
0x000C	GICC_IAR	RO	0x000003FF	I e Ac ed e Re e
0x0010	GICC_EOIR	WO	-	E d f I e Re e
0x0014	GICC_RPR	RO	0x000000FF	R P Re e
0x0018	GICC_HPPIR	RO	0x000003FF	H e P Pe d I e Re e
0x001C	GICC_ABPR[b]	RW	0x0000000x[a]	A a ed B a P Re e
0x0020	GICC_AIAR[c]	RO	0x000003FF	A a ed I e Ac ed e Re e
0x0024	GICC_AEOIR[c]	WO	-	A a ed E d f I e Re e
0x0028	GICC_AHPPIR[c]	RO	0x000003FF	A a ed H e P Pe d I e Re e
0x002C-0x003C	-	-	-	Re e ed
0x0040-0x00CF	-	-	-	IMPLEMENTATION DEFINED e e
0x00D0-0x00DC	GICC_APR [c]	RW	0x00000000	Ac e P e Re e
0x00E0-0x00EC	GICC_NSAPR [c]	RW	0x00000000	N - ec e Ac e P e Re e
0x00ED-0x00F8	-	-	-	Re e ed
0x00FC	GICC_IIDR	RO	IMPLEMENTATION DEFINED	CPU I e face Ide f ca Re e
0x1000	GICC_DIR[c]	WO	-	Deac a e I e Re e

a. See e e e de c f e f a .
b. P e e GIC 1 f e GIC e e e GIC Sec E e . A a e e GIC 2.
c. GIC 2 .

4.1.4 GIC register access

A e e 32-b d acce e e acce e def ed Tab e 4-1 a e 4-75 a d Tab e 4-2 a e 4-76.

I add , e GICD_IPRIORITYR , GICD_ITARGETSR , GICD_CPENDSGIR , a d GICD_SPENDSGIR e e b e acce e .

W e e a a f d e e acce e a e e ed IMPLEMENTATION DEFINED.

Note

I e GIC a c ec e, a e e a a e a f d-acce b e b e-acce b e e a e e d a e de de .

If e GIC e e e GIC Sec E e e e affec e e acce e a f :
e e e a e ba ed, ee *R*
e e e a e acce b e Sec e acce e
a , e GIC c d f e a e f e e e .

F e f a ee *E* *GIC S* *E* a e 4-80.

Register banking

Re e ba efe d ec e fa e e a e a e add e . T e e e fa e e acce de e e c c f e e e add e ed. T e GIC ba e e e f ca e :

If e GIC e e e Sec E e , e e e a e ba ed de e a a e Sec e a d N - ec e c e f e e e . T e Sec e a d N - ec e e e b a e ca d ffe . A Sec e acce e e e add e acce e e Sec e c f e e e , a d a N - ec e acce acce e e N - ec e c . See *E* *GIC S* *E* a e 4-80 f e f a .

If e GIC e e ed a a f a ce e :

G S e e e a e ba ed de a e a a e c f eac c ec ed ce . T e e c de e e e a c a ed PPI a d SGI , a d e GICD_NSACR , e e e ed.

G T e GIC e e e CPU e face e e de e de f eac CPU e face, a d eac c ec ed ce acce e e e e e f e e face c ec .

4.1.5 Enabling and disabling the Distributor and CPU interfaces

T ec de c be e ab e a d d ab e e D b a d CPU e face , a d e d ffe e ce be a a e e a a e . I de c be :

I

I a e 4-79.

Implementations that support interrupt grouping

I e e e a GIC 2 e e a a d GIC 1 e e a a c de e GIC Sec E e ,

I a GIC a e :

e GICD_CTLR.E ab eG 0 b c e f a d f G 0 e f e D b e CPU e face

e GICD_CTLR.E ab eG 1 b c e f a d f G 1 e f e D b e CPU e face

e GICC_CTLR.E ab eG 0 b c e a f G 0 e b e CPU e face e ce

e GICC_CTLR.E ab eG 1 b c e a f G 1 e b e CPU e face e ce .

F e D b :

If e GICD_CTLR.E ab eG 0 a d GICD_CTLR.E ab eG 1 b a e b 0:

G e D b d e f a d e d e e CPU e face

G IMPLEMENTATION DEFINED e e a ed e- e ed e a e e e e e d a e.

G ead f GICC_IAR, GICC_AIAR, GICC_HPPIR, GICC_AHPPIR e a e ID

G f a e ca ead e e D b e e

G IMPLEMENTATION DEFINED e e SGI ca be e e d GICD_SGIR

If e e , b b , f e GICD_CTLR.E ab eG 0 a d GICD_CTLR.E ab eG 1 b e 1, a d e e e d e e d ab ed , e D b d e f a d a e d e e CPU e face . A IMPLEMENTATION DEFINED, a e e f ca e :

G GICD_CTLR.E ab eG 0 e 0 a d GICD_CTLR.E ab eG 1 e 1, a d e e e d e 0

G GICD_CTLR.E ab eG 0 e 1 a d GICD_CTLR.E ab eG 1 e 0, a d e e e d e 1.

I a e e a a c de e GIC Sec E e , ea a , ca e e e e a e G 1 e a e a eG 0 e , b e f N - ec e f a e de e ce Sec e f a e, b c ea e GICD_CTLR.E ab eG 1 b . T e e , ARM ec e d a a G 0 e a e a ed a e a a G 1 e .

I add , e e Sec e f a e f de e ce N - ec e f a e, Sec e f a e e e a e GICD_CTLR.E ab eG 1 e 1, e e GICD_CTLR.E ab eG 0 a e 1, a e e a e e d G 0 e .

See *R* a e 3-56 f e f a .

F a CPU e face, e GICC_CTLR.Ac C == 0:

W e GICC_CTLR.E ab eG 0 == 0

G G 0 e f a ded f e D b a e a ed e ce

G a ead f GICC_IAR e a e ID

W e GICC_CTLR.E ab eG 0 == 1, G 0 e f a ded f e D b a e a ed e ce .

W e GICC_CTLR.E ab eG 1 == 0

G G 1 e f a ded f e D b a e a ed e ce

G a ead f GICC_AIAR e a e ID

W e GICC_CTLR.E ab eG 1 == 1, G 1 e f a ded f e D b a e a ed e ce

f e e GICC_CTLR.E ab eG 0 GICC_CTLR.E ab eG 1 e 0, a d e e a e d e f ff c e e d ab ed , IMPLEMENTATION DEFINED e e a ead f GICC_HPPIR e e ID f a e , a e ID.

F a CPU e face, e GICC_CTLR.Ac C == 1:

W e GICC_CTLR.E ab eG 1 == 0, a N - ec e ead f GICC_IAR e a e ID

W e GICC_CTLR.E ab eG 0 == 0:

G f GICC_CTLR.E ab eG 1 == 0, a Sec e ead f GICC_AIAR e a e ID

G f GICC_CTLR.E ab eG 1 == 1, G 0 e a e ed a d GICC_IAR be a e a GICC_AIAR

W e GICC_CTLR.E ab eG 1 == 0, a Sec e ead f GICC_AIAR a a e a e ID

f e e GICC_CTLR.E ab eG 0 GICC_CTLR.E ab eG 1 e 0, a d e e a e d e f ff c e e d ab ed , IMPLEMENTATION DEFINED e e a ead f GICC_HPPIR e e ID f a e , a e ID.

Note

ARM de eca e e f GICC_CTLR.Ac C , a d ec e d a f a e de e e GICC_CTLR.Ac C e 0.

Implementations that do not support interrupt grouping

Note

T e e e a a d e a e GIC 1 e e a a d c de e GIC Sec E e .

I a GIC a d e e :

e GICD_CTLR.E ab e b c e f a d f e f e D b e CPU e face

e GICC_CTLR.E ab e b c e a f e b e CPU e face e c ec ed ce .

F e D b :

W e GICD_CTLR.E ab e e 1, e D b f a d e e e d e f eac CPU e face, b ec e a e .

W e GICD_CTLR.E ab e e 0:

G e D b d e f a d e d e e CPU e face

G IMPLEMENTATION DEFINED e e a ed e- e ed e a e e e e e d a e.

G ead f GICC_IAR, GICC_AIAR, GICC_HPPIR, GICC_AHPPIR e a e ID

G f a e ca ead e e D b e e

G IMPLEMENTATION DEFINED e e SGI ca be e e d GICD_SGIR.

F a CPU e face:

W e GICC_CTLR.E ab e e 1, e e e d e f a ded f e D b e CPU e face a ed e c ec ed ce

W e GICC_CTLR.E ab e e 0:

G a e d e f a ded f e D b a e a ed e ce

G f a e ca ead e e CPU e face e e

G a ead f e GICC_IAR e a e ID

G f e D b f a d a e e CPU e face, a e e face ca a beca e GICC_CTLR.E ab e e 0, IMPLEMENTATION DEFINED e e a ead f GICC_HPPIR e e ID f a e , a e ID.

Note

T e E ab eG 1 b e N - ec e c e f e GICD_CTLR a d GICC_CTLR e e a e c ea ed 0 e e . T ea a f a e ca a e D b a d CPU e face e e bef e e ab e GIC.

See *D C R , GICD CTLR* a e 4-85 a d *CP I C R , GICC CTLR* a e 4-125 f e f a .

4.2 Effect of the GIC Security Extensions on the programmers' model

Note

F a e e f e GIC Sec E e , ee *S E* a e 1-16.

Table 4-3 Registers implemented differently when the GIC includes the GIC Security Extensions (continued)

Register	Type	Description	Effect
GICD_SGIR	C	S f a e Ge e a ed I e Re e	Add e NSATT b
GICC_CTLR	Ba ed	CPU I e face C Re e	Re e ba ed[a]
GICC_BPR	Ba ed	B a P Re e	Re e ba ed[a]
GICC_ABPR	Sec e	A a ed B a P Re e	Re e Sec e
GICC_AIAR	Sec e	A a ed I e Ac ed e Re e	Re e Sec e
GICC_AEOIR	Sec e	A a ed E d f I e Re e	Re e Sec e
GICC_AHPPIR	Sec e	A a ed H e P Pe d I e Re e	Re e Sec e
GICC_NSAPR	Sec e	N - ec e Ac e P e Re e	Re e Sec e

a. F e f a , ee *R* a e 4-77.

T e f ec e e f a ab e effec f e GIC Sec E e e GIC a e de :
N - *G 0*
C a e 4-82.

4.2.1 Non-secure access to register fields for Group 0 interrupt priorities

P *S* *N - GIC* a e 1-20 de def f Sec e f a e a d Sec e a d N - ec e acce e .

T e GIC Sec E e e e f G 0 e a Sec e e , a d G 1 e a N - ec e e . T ea a e e e f ed a c a ed G 0 e a e RAZ/WI N - ec e acce e , a d add :

Non-secure access to a priority field in the GICD_IPRIORITYRn

If e f ed c e d a G 1 e , e acce e a e a def ed b e N - ec e e f e , ee *S* *GIC S E* a e 3-53.

Non-secure access to the GICC_PMR and GICC_RPR

If e c e a a e e a e `0x00-0x7F`:
- G a ead acce e e a e `0x00`
- G e GIC e a e acce e GICC_PMR.

If e c e a a e e a e `0x80-0xFF`:
- G A ead acce e e N - ec e e f e c e a e.
- G A e acce e GICC_PMR cceed , ba ed e N - ec e e f e a a e e e e e . T ea a N - ec e e ca e a a a e e a e `0x00-0x7F`.

T e e d c de *T* *GIC S E* a e 3-66 de c be acce e e GICD_IPRIORITYR , GICC_PMR, a d GICC_RPR e e GIC e e e Sec E e .

4.2.2 Configuration lockdown

A GIC e e a a c de e GIC Sec E e ca e e c f a c d . T de a c a a e e ca a e e e acce :

e e e f e d c ac f ed a e f SPI , e e SPI a e c f ed a G 0 e

e c f a e e .

W e e c a a e ed, e affec ed e e f e d a d e e a e de c bed a be .

L c d c ed b a ac e HIGH d ab e a, **CFGSDISABLE**. T a , e e a e **CFGSDISABLE** HIGH d ab e e acce e e e f e d a d e e .

T e SPI a ca be c ed d a e ca ed *SPI* (LSPI). T e be f LSPI IMPLEMENTATION DEFINED, be ee 0 a d 31:

If e GIC a LSPI e e f b e LSPI a I e ID 32

T e GICD_TYPER.LSPI f e d def e e a be f LSPI . If GICD_TYPER.LSPI ea e a 0 e e b e LSPI a e e ID 32 (31+(GICD_TYPER.LSPI)).

Note

GICD_TYPER.LSPI def e e a e f b e LSPI . T e GIC a e e a e.

If GICD_TYPER.LSPI 0 c d ed. T ea f a e ca c d a c e e f e GIC d e e e a LSPI .

W e e SPI c f e d a d c f a e e a e c e d d , e GIC e e e acce e :

T e E ab e G 0 b f e Sec e c f GICD_CTLR.

T e f b e Sec e c f GICC_CTLR:

G EOI deS
G IRQB D G 0
G FIQB D G 0
G CBPR
G FIQE
G Ac C
G E ab e G 0

See *CP I C R , GICC CTLR* a e 4-125.

F e d e GICD_ISENABLER , GICD_ICENABLER , GICD_ISPENDR , GICD_ICPENDR , GICD_ISACTIVER , GICD_ICACTIVER , GICD_IPRIORITYR , GICD_ITARGETSR , a d GICD_ICFGR e e a c e d L c ab e SPI a a e c f ed a G 0:

F e d e GICD_IGROUPR e e a c e d c ab e SPI a a e c f ed a G 0. If a c ab e SPI ec f ed f G 1 G 0 e **CFGSDISABLE** e a HIGH, e GIC e e a e e GICD_IGROUPR f e d a c e d a SPI, a d e SPI bec e c ed.

T e GIC e a e a c e d d e e e e f e d.

Note

ARM ec e d a , d e e b ce , e e ead e GICD_TYPER.LSPI f e d f d e be f c ab e SPI , a e e e a d e e f e d a ca be c e d d , a d e a e **CFGSDISABLE** HIGH. N a , ea a e Sec e b e e ce a f a f e e e a a e Sec e c f a c de.

ARM ec e d a e **CFGSDISABLE** f a e ed HIGH d e e b
ce , e e e e **CFGSDISABLE** ca be dea e ed e ce d a ce e -d
e e e e ce.

4.2.3 Effect of the Virtualization Extensions on the programmers' model

T e GIC V a a E e add e GIC a e face c e e a d e a CPU e face
e e e a e ' de . See C a e 5 *GIC S* f e f a .

4.3 Distributor register descriptions

T e f ec de c be e D b e e :

D C R , GICD CTLR a e 4-85
I C T R , GICD T PER a e 4-88
D I I R , GICD IIDR a e 4-90
I G R , GICD IGRO PR a e 4-91
I S -E R , GICD ISENABLER a e 4-93
I C -E R , GICD ICENABLER a e 4-95
I S -P R , GICD ISPENDR a e 4-97
I C -P R , GICD ICPENDR a e 4-99
I S -A R , GICD ISACTI ER a e 4-102
I C -A R , GICD ICACTI ER a e 4-103
I P R , GICD IPRIORIT R a e 4-104
I P T R , GICD ITARGETSR a e 4-106
I C R , GICD ICFGR a e 4-109
N - A C R , GICD NSACR a e 4-111
S G I R , GICD SGIR a e 4-113
SGI C -P R , GICD CPENDSGIR a e 4-115
SGI S -P R , GICD SPENDSGIR a e 4-117
I a e 4-119.

See *D* a e 4-74 f add e ff e a d e e f a f e e e e .

4.3.1 Distributor Control Register, GICD_CTLR

T e GICD_CTLR c a ac e c a e:

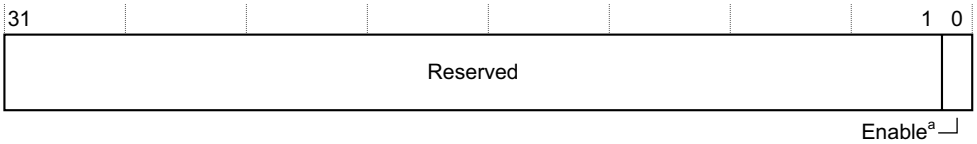
Purpose E ab e e f a d f e d e f e D b e CPU e face .

Usage constraints If e GIC e e e Sec E e c f a c d , e e ca c d e Sec e GICD_CTLR, ee *C* a e 4-82.

Configurations T e e a a ab e a c f a f e GIC. If e GIC e e e Sec E e , e e ba ed, ee *R* a e 4-77.

Attributes See e e e a Tab e 4-1 a e 4-75.

F e 4-1 a d Tab e 4-4 e GICD_CTLR b a e f :
a GIC 1 e e a a d e c de e GIC Sec E e
e N - ec e c f e e e a e e a a c de e GIC Sec E e .



[a] Bit name is IMPLEMENTATION DEFINED in an implementation that includes the Security Extensions

Figure 4-1 GICD_CTLR bit assignments, GICv1 without Security Extensions or Non-secure

Table 4-4 GICD_CTLR bit assignments, GICv1 without Security Extensions or Non-secure

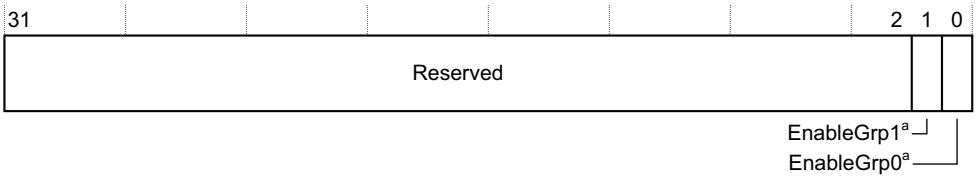
Bits	Name	Function
[31:1]	-	Re e ed.
[0]	E ab e[a]	G ba e ab e f f a d e d e f e D b e CPU e face . I e N - ec e c f e e e a e e a a c de e Sec E e , b c e f a d f G 1 e : **0** e f a ded. **1** e f a ded, b ec e e . See *E D CP* a e 4-77 f e f a ab b .

a. B a e IMPLEMENTATION DEFINED a GIC 1 e e a a c de e Sec E e .

F e 4-2 a d Tab e 4-5 a e 4-86 e GICD_CTLR b a e f :

A GIC 2 e e a . If e e e a c de e Sec E e e e a e a e Sec e c f e e e.

T e Sec e c f e e e a GIC 1 e e a a c de e Sec E e .



[a] In a GICv1 implementation that includes the Security Extensions:
- Bit[0] is named Enable
- Bit[1] is IMPLEMENTATION DEFINED.

Figure 4-2 GICD_CTLR bit assignments, GICv2, and GICv1 Secure copy

Table 4-5 GICD_CTLR bit assignments, GICv2, and GICv1 Secure copy

Bits	Name	Function
[31:2]	-	Re e ed.
[1]	E ab eG 1	G ba e ab e f f a d e d G 1 e f e D b e CPU e face : **0** G 1 e f a ded. **1** G 1 e f a ded, b ec e e . **Note** I a GIC 1 e e a a c de e Sec E e : W e e b e e ed, a d e b a e f e e ed, IMPLEMENTATION DEFINED. If e e ed e b e e ed. W e e b e e ed, a a a f b [0] f e N - ec e c f e e e .
[0]	E ab eG 0	G ba e ab e f f a d e d G 0 e f e D b e CPU e face : **0** G 0 e f a ded. **1** G 0 e f a ded, b ec e e .

W e a f e D b ba e ab e b a e e 0, d ab e D b f c , e GIC e e ead a d e e a e a . T ea f a e ca c a e e a e f PPI a d SPI bef e e-e ab e D b . F e a e, f a e ca :

Ma e a e e d b e c e d GICD_ISPENDR .

Re e e ac e a e f a e b e c e d GICC_EOIR GICC_AEOIR.

Note

Se a D b ba e ab e b 0 d ab e f a d f e e CPU e face . I add :

W e f a d f e d e d ab ed f G 0 G 1 e , IMPLEMENTATION DEFINED e e a ed e- e ed e a e a ed e- e ed e a d ab ed e e d a e.

I GIC 2, f a e ca a a e SGI e d a e e I e Se -Pe d Re e , GICD_ISPENDR a d I e C ea -Pe d Re e , GICD_ICPENDR . H e e , GIC 1, e GIC c ea e e d a e f a SGI e e SGI bec e ac e, a d e ef e f a e ca c ea e e d a e f a SGI.

I GIC 2, f a e ca a a e e ac e a e e I e Se -Ac e Re e , GICD_ISACTIVER a d e I e C ea -Ac e Re e , GICD_ICACTIVER .

If e f a d f e f e d ab ed, a d e e e d e e d ab ed :

I GIC 1, IMPLEMENTATION DEFINED e e e D b f a d a e d e f S ff c e f e e , e CPU e face .

I GIC 2, e D b d e f a d a e , f e e , e CPU e face .

W e e GICD_CTLR. E ab eG 1, E ab eG 0 e ea e D b d e f a d a e d e e CPU e face , a ead f a GICC_IAR GICC_AIAR e e e a e ID.

Note

I e a e, b def , a c e e a d e e a e a ea a b f e e da e. S f a e c de a e c a e a c a ed e e f e d e a CPU e face d a .

4.3.2 Interrupt Controller Type Register, GICD_TYPER

T e GICD_TYPER c a ac e c a e:

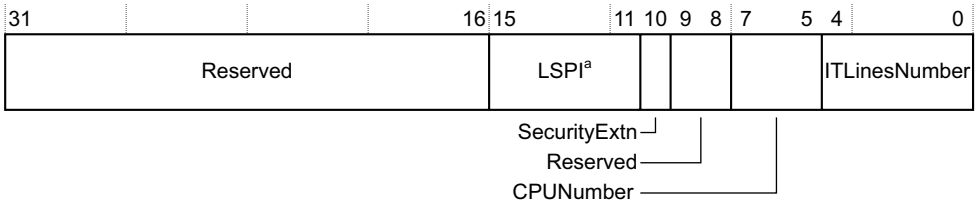
Purpose P de f a ab e c f a f e GIC. I d ca e :
e e e GIC e e e Sec E e
e a be f e ID a e GIC
e be f CPU e face e e ed
f e GIC e e e Sec E e , e a be f e e ed *L S P I* (LSPI).

Usage constraints N a e c a .

Configurations T e e a a ab e a c f a f e GIC. If e GIC e e e Sec E e e e C .

Attributes See e e e a Tab e 4-1 a e 4-75.

F e 4-3 e GICD_TYPER b a e .



a Implemented only if the GIC implements the Security Extensions, Reserved otherwise

Figure 4-3 GICD_TYPER bit assignments

Tab e 4-6 e GICD_TYPER b a e .

Table 4-6 GICD_TYPER bit assignments

Bits	Name	Function
[31:16]	-	Re e ed.
[15:11]	LSPI	If e GIC e e e Sec E e , e a e f f e d e a be f e e ed c ab e SPI , f 0 (`0b00000`) 31 (`0b11111`), ee *C* a e 4-82. If f e d `0b00000` e e GIC d e e e c f a c d . If e GIC d e e e e Sec E e , f e d e e ed.
[10]	Sec E	I d ca e e e e GIC e e e Sec E e . **0** Sec E e e e ed. **1** Sec E e e e ed.

Table 4-6 GICD_TYPER bit assignments (continued)

Bits	Name	Function
[9:8]	-	Re e ed.
[7:5]	CPUN be	I d ca e e be f e e ed CPU e face . T e be f e e ed CPU e face e e a e a e f f e d, f e a e f f e d 0b011, e e a e f CPU e face . If e GIC e e e V a a E e , a e be f a CPU e face .
[4:0]	ITL e N be	I d ca e e a be f e a e GIC . If ITL e N be =N, e a be f e 32(N+1). T e e ID a e G f 0 (be f ID 1). F e a e: 0b00011 U 128 e e , e ID 0-127. T e a be f e 1020 (0b11111). See e e ec f e f a . Re a d e f e a e f e ID def ed b f e d, e ID 1020-1023 a e e e ed f ec a e , ee *S* a e 3-43 a d *I ID* a e 2-24.

T e ITL e N be f e d d ca e e a be f SPI a e GIC . T a e de e e e be f e e ed e e e , a , e be f a ce f ef e e :

- GICD_IGROUPR
- GICD_ISENABLER
- GICD_ICENABLER
- GICD_ISPENDR
- GICD_ICPENDR
- GICD_ISACTIVER
- GICD_IPRIORITYR
- GICD_ITARGETSR
- GICD_ICFGR .

T e GIC a c ec e d e e e a GIC a c a e f SPI e ID , a d e ed SPI e ID a e e be -c . S f a e c ec c SPI e ID a e ed, e a a e d ca ed b e ITL e N be f e d, ee *I* a e 3-35.

4.3.3 Distributor Implementer Identification Register, GICD_IIDR

T e GICD_IIDR c a ac e c a e:

Purpose P de f a ab e e e e a d e f e D b .

Usage constraints N a e c a .

Configurations T e e a a ab e a c f a f e GIC. If e GIC e e e Sec E e e e C .

Attributes See e e e a Tab e 4-1 a e 4-75.

F e 4-4 e GICD_IIDR b a e .

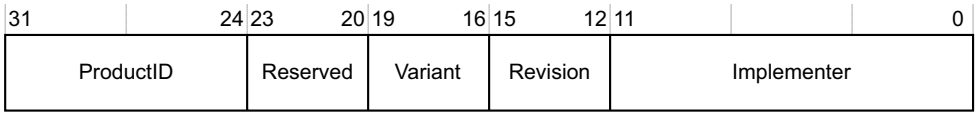


Figure 4-4 GICD_IIDR bit assignments

Tab e 4-7 e GICD_IIDR b a e .

Table 4-7 GICD_IIDR bit assignments

Bits	Name	Function
[31:24]	P d c ID	A IMPLEMENTATION DEFINED d c de f e .
[23:20]	-	Re e ed.
[19:16]	Va a	A IMPLEMENTATION DEFINED a a be . T ca , f e d ed d d c a a , a e fa d c .[a]
[15:12]	Re	A IMPLEMENTATION DEFINED e be . T ca , f e d ed d e fa d c .[a]
[11:0]	I e e e	C a e JEP106 c de f e c a a e e ed e GIC D b : **Bits [11:8]** T e JEP106 c a c de f e e e e . F a ARM e e a , f e d `0x4`. **Bits [7]** A a 0. **Bits [6:0]** T e JEP106 de c de f e e e e . F a ARM e e a , b [7:0] a e `0x3B`.

a. T f e d ed d be ee GIC 1 a d GIC 2 e e a .

4.3.4 Interrupt Group Registers, GICD_IGROUPRn

T e GICD_IGROUPR c a ac e c a e:

Purpose T e GICD_IGROUPR e e de a a b f eac e ed b e GIC. Eac b c e e ec e d e G 0 G 1.

Usage constraints I e e a a c de e GIC Sec E e , acce b e b Sec e acce e . T e e e add e e a e RAZ/WI N - ec e acce e .

A e e b c e d a e e ed e RAZ/WI.

If e GIC e e c f a c d , e e ca c d e a b f c ab e SPI a a e c f ed a G 0, ee *C* a e 4-82.

Configurations I GIC 1, e e ed f e GIC e e e Sec E e . If a GIC 1 e e a d e c de e Sec E e e GICD_IGROUPR add e e a e RAZ/WI.

Note

T ca , e ed a ce a e e e ARM Sec E e , G 0 e a e Sec e e , a d G 1 e a e N - ec e e , ee *S E* a e 1-16 f e f a .

I GIC 2, e e e a e a a e e ed.

T e be f e e ed GICD_IGROUPR e e (GICD_TYPER.ITL e N be + 1). T e e e ed GICD_IGROUPR e e be a d f GICD_IGROUPR0. If e GIC e e e Sec E e , e e a e Sec e e e .

I a ce e e a , GICD_IGROUPR0 ba ed f eac c ec ed ce . T e e d e a b f e 0-31.

Attributes See e e e a Tab e 4-1 a e 4-75, a d *GICD IGRO PR0* a e 4-92.

F e 4-5 e GICD_IGROUPR b a e .

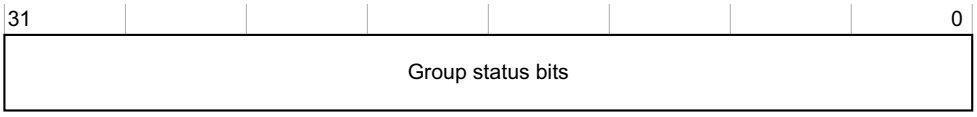


Figure 4-5 GICD_IGROUPR bit assignments

Tab e 4-8 e GICD_IGROUPR b a e .

Table 4-8 GICD_IGROUPR bit assignments

Bits	Name	Function
[31:0]	G a b	F eac b : **0** T e c e d e G 0. **1** T e c e d e G 1.

Note

4.3.5 Interrupt Set-Enable Registers, GICD_ISENABLERn

T e GICD_ISENABLER c a ac e c a e:

Purpose T e GICD_ISENABLER de a Se -e ab e b f eac e ed b e GIC. W 1 a Se -e ab e b e ab e f a d f ec e d e f e D b e CPU e face . Read a b de f e e e e e ab ed.

Usage constraints A e e b c e d a e e ed e RAZ/WI.

If e GIC e e e Sec E e :

a e e b a c e d a G 0 e RAZ/WI N - ec e acce e

f e GIC e e c f a c d , e e ca c d e Se -e ab e b f e c ab e SPI a a e c f ed a G 0, ee *C* a e 4-82.

W e e e e ed SGI a e e a e e ab ed, ca be e ab ed a d d ab ed b e GICD_ISENABLER0 a d GICD_ICENABLER0, IMPLEMENTATION DEFINED.

Configurations T e e e e a e a a ab e a c f a f e GIC. If e GIC e e e Sec E e e e e e a e C .

T e be f e e ed GICD_ISENABLER (GICD_TYPER.ITL e N be +1). T e e e ed GICD_ISENABLER be a d f GICD_ISENABLER0.

I a ce e e a , GICD_ISENABLER0 ba ed f eac c ec ed ce . T e e d e Se -e ab e b f e 0-31.

Attributes See e e e a Tab e 4-1 a e 4-75.

F e 4-6 e GICD_ISENABLER b a e .

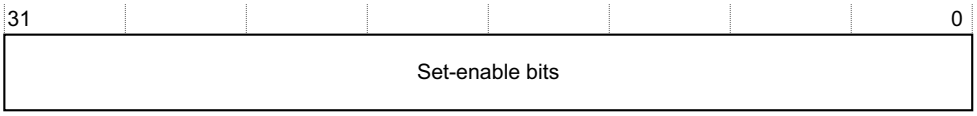


Figure 4-6 GICD_ISENABLER bit assignments

Tab e 4-9 e GICD_ISENABLER b a e .

Table 4-9 GICD_ISENABLER bit assignments

Bits	Name	Function
[31:0]	Se -e ab e b	F SPI a d PPI , eac b c e f a d f ec e d e f e D b e CPU e face : **Reads** **0** F a d f ec e d e d ab ed. **1** F a d f ec e d e e ab ed. **Writes** **0** Ha effec . **1** E ab e e f a d f ec e d e . Af e a e f 1 ab , a b e e ead f e b e e a e 1. F SGI e be a f e b ead a d e IMPLEMENTATION DEFINED.

F e ID , e DIV a d MOD a e e e d a d d e a :
e c e d GICD_ISENABLER be , , e b = DIV 32
e ff e f e e ed GICD_ISENABLER (0x100 + (4*))
e b be f e e ed Se -e ab e b e e MOD 32.

A a - , a d af e a e e , a ce ca e e e d c e c e e a e ID e GIC . If e ce a d e GIC b e e e Sec E e d f e Sec e e f ea a ab e e , a d N - ec e f a e e ce d d c e af e e Sec e f a e a c f ed e a G 0 (Sec e) a d G 1 (N - ec e). F e f a ee *I* a e 3-35.

Note

D ab a e d ab e e f a d f e e f e D b a CPU e face. I d e e e e f c a a e, f e a e bec e d , ac e a d e d f a ead ac e.

4.3.6 Interrupt Clear-Enable Registers, GICD_ICENABLERn

T e GICD_ICENABLER c a ac e c a e:

Purpose T e GICD_ICENABLER de a C ea -e ab e b f eac e ed b e GIC. W 1 a C ea -e ab e b d ab e f a d f ec e d e f e D b e CPU e face . Read a b de f e e e e e ab ed.

Usage constraints A e e b c e d a e e ed e RAZ/WI.

If e GIC e e e Sec E e :

a e e b a c e d a G 0 e RAZ/WI N - ec e acce e

f e GIC e e c f a c d , e e ca c d e C ea -e ab e b f e c ab e SPI a a e c f ed a G 0, ee *C* a e 4-82.

W e e e e ed SGI a e e a e e ab ed, ca be e ab ed a d d ab ed b e GICD_ISENABLER0 a d GICD_ICENABLER0, IMPLEMENTATION DEFINED.

Configurations T e e e e a e a a ab e a c f a f e GIC. If e GIC e e e Sec E e e e e e a e C .

T e be f e e ed GICD_ICENABLER (GICD_TYPER.ITL e N be +1). T e e e ed GICD_ICENABLER be a d f GICD_ICENABLER0.

I a ce e e a , GICD_ICENABLER0 ba ed f eac c ec ed ce . T e e d e C ea -e ab e b f e 0-31.

Attributes See e e e a Tab e 4-1 a e 4-75.

F e 4-7 e GICD_ICENABLER b a e .

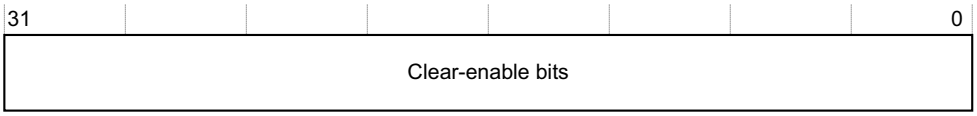


Figure 4-7 GICD_ICENABLER bit assignments

Tab e 4-10 e GICD_ICENABLER b a e .

Table 4-10 GICD_ICENABLER bit assignments

Bits	Name	Function
[31:0]	C ea -e ab e b	F SPI a d PPI , eac b c e f a d f ec e d e f e D b e CPU e face : **Reads** **0** F a d f ec e d e d ab ed. **1** F a d f ec e d e e ab ed. **Writes** **0** Ha effec . **1** D ab e e f a d f ec e d e . Af e a e f 1 a b , a b e e ead f e b e e a e 0. F SGI e be a f e b ead a d e IMPLEMENTATION DEFINED.

F e ID , e DIV a d MOD a e e e d a d d e a :
e c e d GICD_ICENABLER be , , e b = DIV 32
e ff e f e e ed GICD_ICENABLER (0x180 + (4*))
e b be f e e ed C ea -e ab e b e e MOD 32.

4.3.7 Interrupt Set-Pending Registers, GICD_ISPENDRn

T e GICD_ISPENDR c a ac e c a e:

Purpose T e GICD_ISPENDR de a Se - e d b f eac e ed b e GIC. W 1 a Se - e d b e e a f ec e d e e a e e d . Read a b de f e e e e e d .

Usage constraints A e e b c e d a e e ed e RAZ/WI.

If e GIC e e e Sec E e :

a e e b a c e d a G 0 e RAZ/WI N - ec e acce e

f e GIC e e c f a c d , e e ca c d e Se - e d b f e c ab e SPI a a e c f ed a G 0, ee *C* a e 4-82.

Se - e d b f SGI a e ead- a d e e .

Configurations T e e e e a e a a ab e a c f a f e GIC. If e GIC e e e Sec E e e e e e a e C .

T e be f e e ed GICD_ISPENDR (GICD_TYPER.ITL e N be +1). T e e e ed GICD_ISPENDR be a d f GICD_ISPENDR0.

I a ce e e a , GICD_ISPENDR0 ba ed f eac c ec ed ce . T e e d e Se - e d b f e 0-31.

Attributes See e e e a Tab e 4-1 a e 4-75.

F e 4-8 e GICD_ISPENDR b a e .

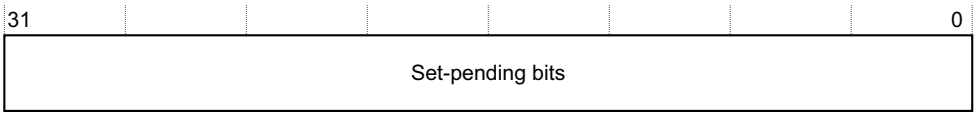


Figure 4-8 GICD_ISPENDR bit assignments

Tab e 4-11 a e 4-98 e GICD_ISPENDR b a e .

Table 4-11 GICD_ISPENDR bit assignments

Bits	Name	Function
[31:0]	Se - e d b	F eac b :
		Reads **0** T ec e d e e d a ce .
		1 F PPI a d SGI , ec e d e e d [a] ce .
		F SPI , ec e d e e d [a] a ea e ce .
		Writes F SPI a d PPI :
		0 Ha effec .
		1 T e effec de e d e e e e ed e- e ed e e - e e:
		Edge-triggered
		C a e e a f ec e d e : e d f a e ac e ac ea d e d f a e ac e.
		Ha effec f e e a ead e d [a].
		Level sensitive
		If ec e d e e d [a], c a e e a f ec e d e : e d f a e ac e ac ea d e d f a e ac e.
		If e e a ead e d [a]: beca e f a e e GICD_ISPENDR, e e a effec beca e ec e d e a a e ed, e e a effec e a f e e , b e e e a e d [a] f e e a dea e ed.
		F e f a ee *C* - a e 4-100.
		F SGI , e e ed. SGI a e e Se -Pe d e e , ee *SGI S -P R , GICD SPENDSGIR* a e 4-117.

a. Pe d e c de e a a e ac ea d e d .

F e ID , e DIV a d MOD a e e e d a d d ea :
ec e d GICD_ISPENDR be , , e b = DIV 32
e ff e f e e ed GICD_ISPENDR (0x200 + (4*))
e b be f e e ed Se - e d b e e MOD 32.

4.3.8 Interrupt Clear-Pending Registers, GICD_ICPENDRn

T e GICD_ICPENDR c a ac e c a e:

Purpose T e GICD_ICPENDR de a C ea - e d b f eac e ed b e GIC. W 1 a C ea - e d b c ea e e d a e f ec e d e e a e . Read a b de f e e e e e e d .

Usage constraints A e e b c e d a e e ed e RAZ/WI.

If e GIC e e e Sec E e :

a e e b a c e d a G 0 e RAZ/WI N - ec e acce e

f e GIC e e c f a c d , e e ca c d e C ea - e d b f e c ab e SPI a a e c f ed a G 0, ee *C* a e 4-82.

C ea - e d b f SGI a e ead- a d e e .

Configurations T e e e e a e a a ab e a c f a f e GIC. If e GIC e e e Sec E e e e e e a e C .

T e be f e e ed GICD_ICPENDR (GICD_TYPER.ITL e N be +1). T e e e ed GICD_ICPENDR be a d f GICD_ICPENDR0.

I a ce e e a , GICD_ICPENDR0 ba ed f eac c ec ed ce . T e e d e C ea - e d b f e 0-31.

Attributes See e e e a Tab e 4-1 a e 4-75.

F e 4-9 e GICD_ICPENDR b a e .

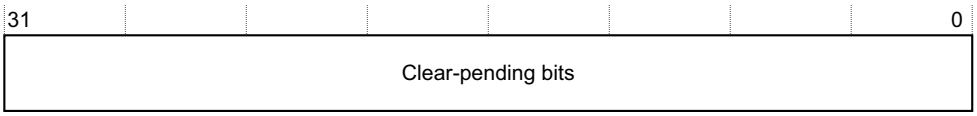


Figure 4-9 GICD_ICPENDR bit assignments

Tab e 4-12 a e 4-100 e GICD_ICPENDR b a e .

Table 4-12 GICD_ICPENDR bit assignments

Bits	Name	Function
[31:0]	C ea - e d b	F eac b : **Reads** **0** T ec e d e e d a ce . **1** F SGI a d PPI , ec e d e e d [a] ce . F SPI , ec e d e e d [a] a ea e ce . **Writes** F SPI a d PPI : **0** Ha effec . **1** T e effec de e d e e e e ed e- e ed e e - e e: **Edge-triggered** C a e e a f ec e d e : ac e f a e e d ac e f a e ac ea d e d . Ha effec f e e e d . **Level-sensitive** If ec e d e e d [a] beca e fa e GICD_ISPENDR , e ec a e e a f e e : ac e f a e e d ac e f a e ac ea d e d . O e e e e e a e d f e e a e a a e ed, ee *C -* F SGI , e e ed. SGI a e e C ea -Pe d e e , ee *SGI C -P R , GICD CPENDSGIR* a e 4-115.

a. Pe d e c de e a a e ac ea d e d .

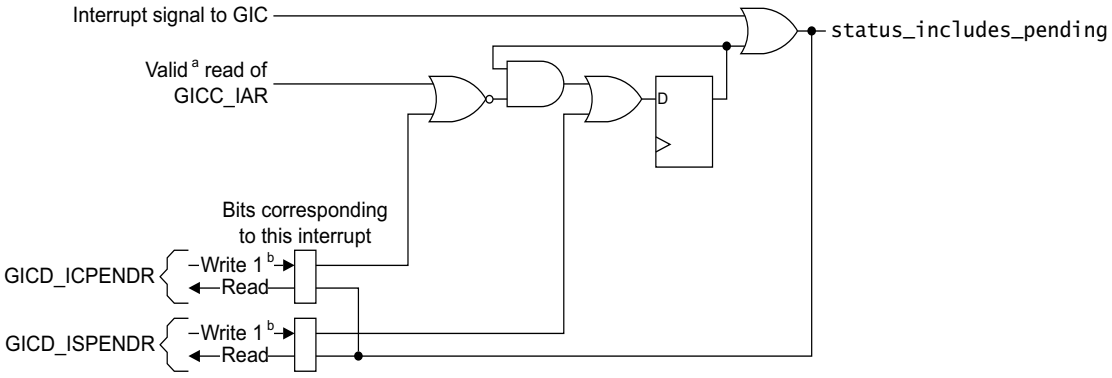
F e ID , e DIV a d MOD a e e e e d a d d e a :
e c e d GICD_ICPENDR be , , e b = DIV 32
e ff e f e e ed GICD_ICPENDR (0x280 + (4*))
e b be f e e ed Se - e d b e e MOD 32.

Control of the pending status of level-sensitive interrupts

T b ec de c be e a fa e a f e e a e f:
e d
ac e a d e d .

F a ed e- e ed e , e c de e d a ac ed e e a e e GICD_ISPENDR e a e f e e a e GIC. H e e, f a e e - e e e , e c de e d a e e :
a c ed a e e GICD_ISPENDR
f e a e f e e a e GIC, a a c .

T ea a e e a f e Se - e d a d C ea - e d e e ec ca ed f e e - e e e . F e 4-10 a e 4-101 e c f e e d a fa e e - e e e . T e ca `status_includes_pending` TRUE e e e a c de e d , a d FALSE e e.



a A read that acknowledges this interrupt
b The register ignores a write of 0

GICC_IAR: Interrupt Acknowledge Register
GICD_ICPENDR: Interrupt Clear-Pending Register
GICD_ISPENDR: Interrupt Set-Pending Register

Figure 4-10 Logic of the pending status of a level-sensitive interrupt

4.3.9 Interrupt Set-Active Registers, GICD_ISACTIVERn

T e GICD_ISACTIVER c a ac e c a e:

Purpose T e GICD_ISACTIVER de a Se -ac e b f eac e a e GIC . W a Se -ac e b Ac a e e c e d e . T e e e a e ed e e e a d e GIC a e.

I GIC 1, e GICD_ISACTIVER e e a e e RO Ac e B Re e , ICDABR .

Usage constraints A e e b c e d a e e ed e RAZ/WI.

If e GIC e e e Sec E e a e e b a c e d a G 0 e RAZ N - ec e acce e .

T e b ead a e f e a f e e ac e ac e a d e d . Read e GICD_ISPENDR GICD_ICPENDR f d e e d a f e e .

Configurations T e e e e a e a a ab e a c f a f e GIC. If e GIC e e e Sec E e e e e e a e C .

T e be f e e ed GICD_ISACTIVER (GICD_TYPER.ITL e N be +1). T e e e ed GICD_ISACTIVER be a d f GICD_ISACTIVER0.

I a ce e e a , GICD_ISACTIVER0 ba ed f eac c ec ed ce . T e e d e Se -ac e b f e 0-31.

T e e e e a e RO GIC 1 a d RW GIC 2.

Attributes See e e e a Tab e 4-1 a e 4-75.

F e 4-11 e GICD_ISACTIVER b a e .

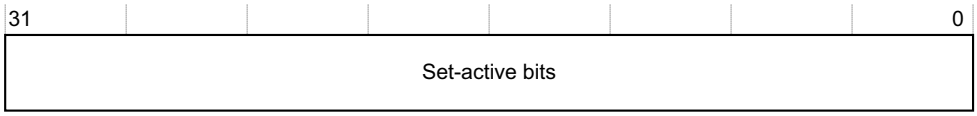


Figure 4-11 GICD_ISACTIVER bit assignments

Tab e 4-13 e GICD_ISACTIVER b a e .

Table 4-13 GICD_ISACTIVER bit assignments

Bits	Name	Function		
[31:0]	Se -ac e b	F eac b :		
		Reads	**0**	T e c e d e ac e[a].
			1	T e c e d e ac e[a].
		Writes	**0**	Ha effec .
			1	Ac a e e c e d e , f a ead ac e. If e e a ead ac e, e e a effec . Af e a e f 1 b , a b e e ead f e b e e a e 1.

a. Ac e e c de e a a e ac e a d e d .

F e ID , e DIV a d MOD a e e e d a d d e a :
e c e d GICD_ISACTIVER be , , e b = DIV 32
e ff e f e e ed GICD_ISACTIVER (0x300 + (4*))
e b be f e e ed Se -ac e b e e MOD 32.

4.3.10 Interrupt Clear-Active Registers, GICD_ICACTIVERn

T e GICD_ICACTIVER c a ac e c a e:

Purpose T e GICD_ICACTIVER de a C ea -ac e b f eac e a e GIC . W a C ea -ac e b Deac a e e c e d e . T e e e a e ed e e e a d e GIC a e.

Usage constraints A e e b c e d a e e ed e RAZ/WI.

If e GIC e e e Sec E e , a e e b a c e d a G 0 e RAZ/WI N - ec e acce e .

Configurations T e e e a e e e GIC 2. T e e e ca a e e e ed GIC 1.

T e be f e e ed GICD_ICACTIVER (GICD_TYPER.ITL e N be +1). T e e e ed GICD_ICACTIVER be a d f GICD_ICACTIVER0.

I a ce e e a , GICD_ICACTIVER0 ba ed f eac c ec ed ce . T e e d e C ea -ac e b f e 0-31.

Attributes See e e e a Tab e 4-1 a e 4-75.

F e 4-12 e GICD_ICACTIVER b a e .

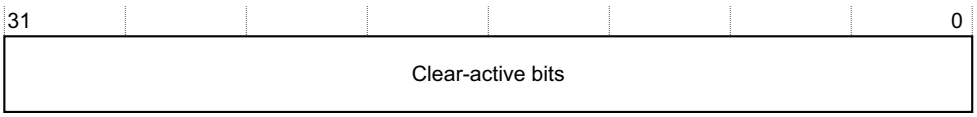


Figure 4-12 GICD_ICACTIVER bit assignments

Tab e 4-14 e GICD_ICACTIVER b a e .

Table 4-14 GICD_ICACTIVER bit assignments

Bits	Name	Function		
[31:0]	C ea -ac e b	F eac b :		
		Reads	**0**	T e c e d e ac e[a].
			1	T e c e d e ac e[a].
		Writes	**0**	Ha effec .
			1	Deac a e e c e d e , f e e ac e. If e e a ead deac a ed, e e a effec . Af e a e f 1 b , a b e e ead f e b e e a e 0.

a. Ac e e c de e a a e ac e a d e d .

F e ID , e DIV a d MOD a e e e d a d d e a :
e c e d GICD_ICACTIVER be , , e b = DIV 32
e ff e f e e ed GICD_ICACTIVER (0x380 + (4*))
e b be f e e ed C ea -ac e b e e MOD 32.

4.3.11 Interrupt Priority Registers, GICD_IPRIORITYRn

T e GICD_IPRIORITYR c a ac e c a e:

Purpose T e GICD_IPRIORITYR de a 8-b f e d f eac e ed b e GIC. T f e d e e f ec e d e .

Usage constraints T e e e e a e b e-acce b e.

A e e f e d c e d a e e e ed e RAZ/WI.

A GIC e e fe e a e b , b e e e a ea b [7:4] f eac f e d. I eac f e d, e e ed b a e RAZ/WI.

If e GIC e e e Sec E e :

a e e f e d a c e d a G 0 e RAZ/WI N - ec e acce e

a N - ec e acce a f e d a c e d a G 1 e be a e a de c bed *S GIC S E* a e 3-53

f e GIC e e c f a c d , e e ca c d e P f e d f e c ab e SPI a a e c f e d a G 0, ee *C* a e 4-82

I IMPLEMENTATION DEFINED e e c a e a e f a f e d c a e e f a ac e e .

Configurations T e e e e a e a a ab e a c f a f e GIC. If e GIC e e e Sec E e e e e e a e C .

T e be f e e ed GICD_IPRIORITYR (8*(GICD_TYPER.ITL e N be +1)). T e e e ed GICD_IPRIORITYR be a d f GICD_IPRIORITYR0.

I a ce e e a , GICD_IPRIORITYR0 GICD_IPRIORITYR7 a e ba ed f eac c ec ed ce . T e e e e d e P f e d f e 0-31.

Attributes See e e e a Tab e 4-1 a e 4-75.

F e 4-13 e GICD_IPRIORITYR b a e .

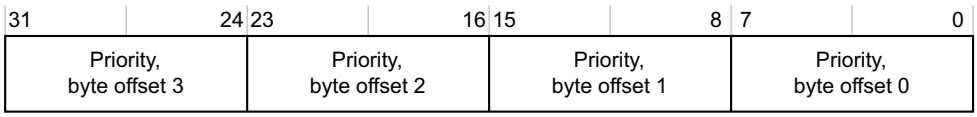


Figure 4-13 GICD_IPRIORITYR bit assignments

Tab e 4-15 e GICD_IPRIORITYR b a e .

Table 4-15 GICD_IPRIORITYR bit assignments

Bits	Name[a]	Function
[31:24]	P , b e ff e 3	Eac f e d d a a e, f a IMPLEMENTATION DEFINED a e. T e e e a e, e ea e e f ec e d e . F e f a ee *I* a e 3-44 a d, f a a e, *I* a e 3-53.
[23:16]	P , b e ff e 2	
[15:8]	P , b e ff e 1	
[7:0]	P , b e ff e 0	

a. Eac f e d d e a e f a e e e . T ec de c be e e ID a e de e e e GICD_IPRIORITYR e e be a d e b e ff e f e f e d a e e.

F e ID , e DIV a d MOD a e e e d a d d e a :
e c e d GICD_IPRIORITYR be , , e b = DIV 4
e ff e f e e ed GICD_IPRIORITYR (0x400 + (4*))
e b e ff e f e e ed P f e d e e MOD 4, e e:

- b e ff e 0 efe e e b [7:0]
- b e ff e 1 efe e e b [15:8]
- b e ff e 2 efe e e b [23:16]
- b e ff e 3 efe e e b [31:24].

T e f e d c de e effec f e GIC Sec E e acce e e e .

```
// PriorityRegRead()
// =================
//

// P_MASK used here to emphasize that the number of valid bits is IMPLEMENTATION DEFINED

bits(8) PriorityRegRead(integer InterruptID)

    read_value = ReadGICD_IPRIORITYR(InterruptID);
    if NS_access then                                       // A non-secure GIC access.
        read_value<7:0> = LSL((read_value AND P_MASK), 1);
        if IsGrp0Int(InterruptID) then
            read_value = '00000000';                        // Can't read a Group 0 priority value
    return(read_value);

// PriorityRegWrite()
// ==================
//

PriorityRegWrite(integer InterruptID, bits(8) value)

    if NS_access then                                       // A non-secure GIC access.
        if !IsGrp0Int(InterruptID) then
            mod_write_val = ('10000000' OR LSR(value,1)) AND P_MASK;
            WriteGICD_IPRIORITYR(InterruptID, mod_write_val);
        else
            IgnoreWriteRequest();
    else                                                    // A secure GIC access.
        mod_write_val = value AND P_MASK;
        WriteGICD_IPRIORITYR(InterruptID, mod_write_val);
```

4.3.12 Interrupt Processor Targets Registers, GICD_ITARGETSRn

T e GICD_ITARGETSR c a ac e c a e:

Purpose T e GICD_ITARGETSR de a 8-b CPU a e f e d f eac e ed b e GIC. T f e d e e f a e ce f e e . T a , d e f CPU e face c e D b f a d e e f a e ed a d a ff c e .

Usage constraints F a ce e e a :

T e e e e a e b e-acce b e.

A e e f e d c e d a e e ed e RAZ/WI.

GICD_ITARGETSR0 GICD_ITARGETSR7 a e ead- , a d eac f e d e a a e a c e d e ce ead e e e .

I IMPLEMENTATION DEFINED c , f a , SPI a e a ca c f ed a d a e. T e CPU a e f e d f c a SPI ead- , a d e a a e a d ca e e CPU a e f e e .

f e GIC e e e Sec E e :

G a e e f e d a c e d a G 0 e RAZ/WI N - ec e acce e

G f e GIC e e c f a c d , e e ca c d e CPU a e f e d f e c ab e SPI a a e c f ed a G 0, ee *C* a e 4-82.

See a *T* *GICD_ITARGETSR* a e 4-108.

Note

I a ce e e a , a e a e e e ce , a d e GICD_ITARGETSR a e RAZ/WI.

Configurations T e e e e a e a a ab e a c f a f e GIC. If e GIC e e e Sec E e e e e e a e C .

T e be f e e ed GICD_ITARGETSR (8*(GICD_TYPER.ITL e N be +1)). T e e e ed GICD_ITARGETSR be a d f GICD_ITARGETSR0.

I a ce e e a , GICD_ITARGETSR0 GICD_ITARGETSR7 a e ba ed f eac c ec ed ce . T e e e e d e CPU a e f e d f e 0-31.

Attributes See e e e a Tab e 4-1 a e 4-75.

F e 4-14 e GICD_ITARGETSR b a e , f a ce e e a .

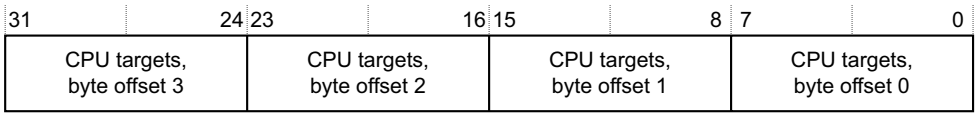


Figure 4-14 GICD_ITARGETSR bit assignments

Tab e 4-16 a e 4-107 e GICD_ITARGETSR b a e , f a ce e e a .

Table 4-16 GICD_ITARGETSR bit assignments

Bits	Name[a]	Function
[31:24]	CPU a e , b e ff e 3	P ce e e be f 0, a d eac b a CPU a e f e d efe e c e d ce , ee Tab e 4-17. F e a e, a a e f 0x3 ea a e Pe d e e ce 0 a d 1. F GICD_ITARGETSR0 GICD_ITARGETSR7, a ead f a CPU a e f e d e e be f e ce e f e ead.
[23:16]	CPU a e , b e ff e 2	
[15:8]	CPU a e , b e ff e 1	
[7:0]	CPU a e , b e ff e 0	

a. Eac f e d d e CPU a e f a e e . T ec de c be e e ID a e de e e e GICD_ITARGETSR e e be a d e b e ff e f e CPU a e f e d a e e.

Tab e 4-17 eac b f a CPU a e f e d a e e e a e f e CPU e face .

Table 4-17 Meaning of CPU targets field bit values

CPU targets field value	Interrupt targets
0bxxxxxxx1	CPU e face 0
0bxxxxxx1x	CPU e face 1
0bxxxxx1xx	CPU e face 2
0bxxxx1xxx	CPU e face 3
0bxxx1xxxx	CPU e face 4
0bxx1xxxxx	CPU e face 5
0bx1xxxxxx	CPU e face 6
0b1xxxxxxx	CPU e face 7

A CPU a e f e d b a c e d a e e e d CPU e face RAZ/WI.

F e ID , e DIV a d MOD a e e e e d a d d e a :
e c e d GICD_ITARGETSR be , , e b = DIV 4
e ff e f e e ed GICD_ITARGETSR (0x800 + (4*))
e b e ff e f e e ed P f e d e e MOD 4, e e:

G b e ff e 0 efe e e b [7:0]
G b e ff e 1 efe e e b [15:8]
G b e ff e 2 efe e e b [23:16]
G b e ff e 3 efe e e b [31:24].

The effect of changes to an GICD_ITARGETSR

S f a e ca e a GICD_ITARGETSR a a e. A c a e a CPU a e f e d a e:

Ha effec a ac e e . T ea a e a CPU e face f a a e d e ca ce a ac e a e f a e a CPU e face.

Ha a effec a e d e . T ea :

- G add a CPU e face e a e f a e d e a e a e e d a CPU e face
- G e a CPU e face f e a e f a e d e e e e e d a e f a e a CPU e face.

Note

T e e a a b f e e e ed f a c a e a e effec .

If a e a e a ac e a d e d , d e c a e e e a e e ac e a c ea ed.

4.3.13 Interrupt Configuration Registers, GICD_ICFGRn

T e GICD_ICFGR c a ac e c a e:

Purpose T e GICD_ICFGR de a 2-b I _c f f e d f eac e ed b e GIC. T f e d de f e e e ec e d e ed e- e ed e e - e e, ee *I* a e 1-18.

Usage constraints F eac ed PPI, IMPLEMENTATION DEFINED e e f a e ca a e c e d I _c f f e d.

F SGI , I _c f f e d a e ead- , ea a GICD_ICFGR0 ead- . F PPI , IMPLEMENTATION DEFINED e e e f ca b f e I _c f f e d a ab e. See Tab e 4-18 a e 4-110 f e f a .

A e e f e d c e d a e e ed e RAZ/WI.

If e GIC e e e Sec E e :

a e e f e d a c e d a G 0 e RAZ/WI N - ec e acce e

f e GIC e e c f a c d , e e ca c d e I _c f f e d f e c ab e SPI a a e c f ed a G 0, ee *C* a e 4-82.

Bef e c a e a e f a a ab e I _c f f e d, f a e d ab e e c e d e , e e GIC be a UNPREDICTABLE.

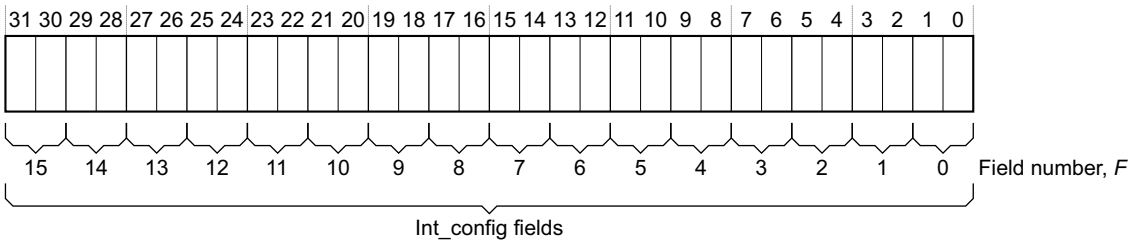
Configurations T e e e e a e a a ab e a c f a f e GIC. If e GIC e e e Sec E e e e e e a e C .

I a ce e e a , f b [1] f e I _c f f e d f a PPI a ab e e GICD_ICFGR1 ba ed f eac c ec ed ce . T e e d e I _c f f e d f e PPI , e 16-31.

T e be f e e ed GICD_ICFGR (2*(GICD_TYPER.ITL e N be +1)). T e e e ed GICD_ICFGR be a d f GICD_ICFGR0.

Attributes See e e e a Tab e 4-1 a e 4-75.

F e 4-15 e GICD_ICFGR b a e .



See the bit assignment table for more information about the properties of each Int_config[1:0] field.

Figure 4-15 GICD_ICFGR bit assignments

Tab e 4-18 a e 4-110 e GICD_ICFGR b a e .

Table 4-18 GICD_ICFGR bit assignments

Bits	Name	Function
[2*F*+1:2*F*]	I _c f , f e d *F*	F I _c f [1], e f ca b , b [2*F*+1], e e c d : **0** C e d e e e - e . **1** C e d e ed e- e ed. I _c f [0], e ea f ca b , b [2*F*], e e ed, b ee Tab e 4-19 f e e c d f b e ea e e a f GIC a c ec e. F SGI : **Int_config[1]** N a ab e, RAO/WI. F PPI a d SPI : **Int_config[1]** F SPI , b a ab e.[a] F PPI IMPLEMENTATION DEFINED e e b a ab e. A ead f b a a e ec ec a e d ca e e e ec e d e e e - e ed e- e ed.

a. If e GIC e e e Sec E e a d e b c e d a G 0 e , RAZ/WI N - ec e acce e . T e a be a f b a c e d G 0 e .

I e e e a f GIC a c ec e bef e e b ca f e GIC 1 A c ec e S ec f ca , e de f a d eac e e a e ca be c f ed b [0] f e c e d I _c f f e d. Tab e 4-19 e e c d f I _c f [0] e e e e a .

Table 4-19 GICD_ICFGR Int_config[0] encoding in some early GIC implementations

Bits	Name	Function
[2*F*]	I _c f [0], f e d *F*	O a GIC e e e a d de f e e a e c f ab e, e e c d f I _c f [0] f PPI a d SPI , : **0** C e d e a d ed e N-N de . **1** C e d e a d ed e 1-N de .

F e ID , e DIV a d MOD a e e e d a d d e a :
e c e d GICD_ICFGR be , , e b = DIV 16
e ff e f e e ed GICD_ICFGR (0xC00 + (4*))
e e ed P f e d e e , *F*, e b *F* = MOD 16, e e f e d 0 efe e e b [1:0], f e d 1 efe b [3:2], f e d 15 a efe b [31:30], ee F e 4-15 a e 4-109.

4.3.14 Non-secure Access Control Registers, GICD_NSACRn

T e GICD_NSACR c a ac e c a e:

Purpose T e GICD_NSACR e ab e Sec e f a e e N - ec e f a e a a c a ce c ea e a d a a e G 0 e . T e de a acce c f eac e e ed e .

Usage constraints T e e e e ca be e e ed f e GIC e e e Sec E e .

T e e e e a e a Sec e e e . If e e ed, ec e d add e ace e e ed.

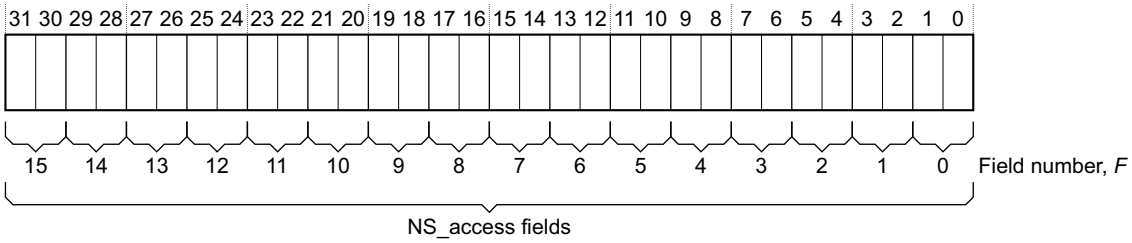
Configurations T e e e e a e e e , a , GIC 2. T e c e d add e ace e e ed GIC 1.

T e c ce f e ec e e ab f N - ec e acce G 0 e a e SGI a d SPI .

GICD_NSACR0 a ba ed e e , a c f e e ce a a a CPU e face a d fea e.

Attributes See e e e a Tab e 4-1 a e 4-75.

F e 4-16 e GICD_NSACR b a e :



See the bit assignment table for more information about the properties of each NS_access[1:0] field.

Figure 4-16 GICD_NSACR bit assignments

Tab e 4-20 e GICD_NSACR b a e :

Table 4-20 GICD_NSACR bit assignments

Bits	Name	Function
[2*F*+1:2*F*]	NS_acce , F e d *F*	If e c e d e d e c f ab e N - ec e acce , e f e d RAZ/WI. O e e, e f e d RW a d c f e e e e f N - ec e acce e e e e e G 0. If e e G 1, f e d e d. T e b e a e f e f e d a e: `0b00` N N - ec e acce e e d f e d a c a e d e c e d e . `0b01` N - ec e e acce e e d f e d a c a e d e c e d e e GICD_ISPENDR e e . A N - ec e e acce GICD_SGIR e e d e e a e a G 0 SGI f e c e d e . `0b10` Add N - ec e e acce e f e d a c a e d e c e d e e GICD_ICPENDR e e . A add N - ec e e ead acce e f e d a c a e d e c e d e e GICD_ISACTIVER a d GICD_ICACTIVER e e . `0b11` Add N - ec e e ead a d e acce e f e d a c a e d e c e d e e GICD_ITARGETSR e e .

T e GICD_NSACR e e d PPI acce e , ea a GICD_NSACR0 b [31:16] a e RAZ/WI.

F e ID , e DIV a d MOD a e e e d a d d e a :
e c e d GICD_NSACR be , , e b = DIV 16
e ff e f e e ed GICD_NSACR (0xE00 + (4*)).

Note

T e add e c e e ed f a Re e acce e -def ed.

4.3.15 Software Generated Interrupt Register, GICD_SGIR

T e GICD_SGIR c a ac e c a e:

Purpose C e e e a f SGI .

Usage constraints I IMPLEMENTATION DEFINED e e e GICD_SGIR a a effec e e f a d f e b D b d ab ed b e GICD_CTLR e .

Configurations T e e a a ab e a c f a f e GIC. If e GIC e e e Sec E e e e C .

T e NSATT f e d, b [15], e e ed

Table 4-21 GICD_SGIR bit assignments (continued)

Bits	Name	Function
[15]	NSATT	I e e ed f e GIC c de e Sec E e . S ec f e e e ed ec a e f e SGI: **0** F a d e SGI ec f ed e SGIINTID f e d a ec f ed CPU e face f e SGI c f ed a G 0 a e face. **1** F a d e SGI ec f ed e SGIINTID f e d a ec f ed CPU e face f e SGI c f ed a G 1 a e face. T f e d ab e b a Sec e acce . A N - ec e e e GICD_SGIR e e a e a SGI f e ec f ed SGI a ed a G 1, e a d e f e a e f b [15] f e e. See *SGI GIC S E* f e f a . **Note** If GIC d e e e e Sec E e , f e d e e ed.
[14:4]	-	Re e ed, SBZ.
[3:0]	SGIINTID	T e I e ID f e SGI f a d e ec f ed CPU e face . T e a e f f e d e I e ID, e a e 0-15, f e a e a a e f `0b0011` ec f e I e ID 3.

a. W e Ta e L F e `0b00`, f e CPUTa e L f e d `0x00` e D b d e f a d e e a CPU e face.

SGI generation when the GIC implements the Security Extensions

If e GIC e e e Sec E e , e e a SGI f a ded a ce ec f ed e e e GICD_SGIR de e d :

e e e e e GICD_SGIR G 0 (Sec e) G 1 (N - ec e)

f a Sec e e e GICD_SGIR, e a e f e GICD_SGIR.NSATT b

e e e ec f ed SGI c f ed a G 0 (Sec e) G 1 (N - ec e) e a e ed ce .

GICD_IGROUPR0 d e ec a e f e SGI , ee e GICD_IGROUPR de c . I a ce e , GICD_IGROUPR0 ba ed f eac c ec ed ce , e e c f e e ec f eac SGI de e de f eac ce . A e e e GICD_SGIR ca a e e a e ce . F eac a e ed ce , e D b de e e e e f a d e SGI e ce .

Tab e 4-22 e ab e f e e e D b f a d a SGI a ec f ed a e CPU e face.

Table 4-22 Truth table for sending an SGI to a target processor

Status of GICD_SGIR write	NSATT value	SGI configuration-3 ((o) 5 (r) -2 (437 (t2-19 (a) 5 (r) -2 (g-76 (e) 5 (t) 5 (p) 5

4.3.16 SGI Clear-Pending Registers, GICD_CPENDSGIRn

T e GICD_CPENDSGIR c a ac e c a e:

Purpose T e GICD_CPENDSGIR de a c ea - e d b f eac ed SGI a d ce ce c b a . W e a ce e a l ac ea - e d b , e e d a e f ec e d SGI f ec e d ce ce e ed, a d e a e e ce e f e e. W a 0 a effec . Read a b de f e e e e SGI e d , f ec e d ce ce , e ead ce .

T e e e e a e ed e e e a d e GIC a e.

Note

I e e e e , a d e GICD_SPENDSGIR e e , a SGI de f ed b e c b a f SGI be a d ce ce .

Usage constraints A e e b c e d a e e ed SGI RAZ/WI.

T e e e e a e b e-acce b e.

If e GIC e e e Sec E e :

a e e b a c e d a G 0 e RAZ/WI N - ec e acce e

f e GIC fe e a e ce , e e b c e d e - e e ed ce a e RAZ/WI.

Note

I a ce e e a , e ce acce e e e ca c a e e SGI e d a e c e d e face. C a e e d a f a SGI f e a e ce d e affec e a f a SGI a e ce .

PPI a d SPI b e e I e C ea -Pe d e e , GICD_ICPENDR .

Configurations T e e e e a e e e GIC 2. T e e e ca a e e e ed GIC 1.

F SGI C ea -Pe d e e a e e e ed. T e e e c a a b f eac f e b e ce ce , f eac f e 16 b e SGI . T a , eac e e c a e c ea - e d b f eac f f SGI .

I a ce e e a , e GICD_CPENDSGIR e e a e ba ed f eac c ec ed ce .

Attributes See e e e a Tab e 4-1 a e 4-75.

F e 4-18 e GICD_CPENDSGIR b a e .

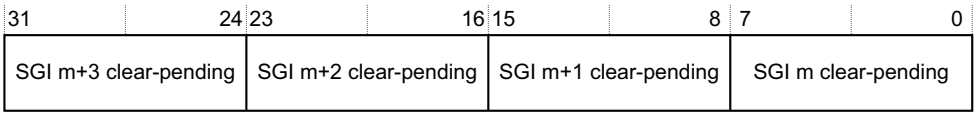


Figure 4-18 GICD_CPENDSGIR bit assignments

Tab e 4-23 e GICD_CPENDSGIR b a e .

Table 4-23 GICD_CPENDSGIRn bit assignments

Bits	Name	Function
[8 +7:8], f =0 3	SGI C ea - e d b	F eac b : **Reads** **0** SGI f e c e d ce e d [a]. **1** SGI f e c e d ce e d [a]. **Writes** **0** Ha effec . **1** Re e e e d a e f SGI f e c e d ce . See e f e e a be ee e SGI be , , e GICD_CPENDSGIR e e be , , a d e f e d be , . **Note** A acce e e a e SGI a a e e ce a e acce .

a. Pe d e c de e a a e ac e a d e d .

F SGI ID , e e a ed b CPU *C* GICD_SGIR, e DIV a d MOD a e e e e d a d d e a :

e c e d GICD_CPENDSGIR e e be , , e b = DIV 4
e ff e f e e ed GICD_CPENDSGIR (0xF10 + (4*));
e SGI C ea - e d f e d ff e , , e b = MOD 4
e e ed b e SGI C ea - e d f e d b C.

4.3.17 SGI Set-Pending Registers, GICD_SPENDSGIRn

T e GICD_SPENDSGIR c a ac e c a e:

Purpose T e GICD_SPENDSGIR e e de a e - e d b f eac ed SGI a d ce ce c b a . W e a ce e a l a e - e d b , e e d a e a ed e c e d SGI f e c e d ce ce . W a 0 a effec . Read a b de f e e e e SGI e d , f e c e d ce ce , e ead ce .

T e e e e a e ed e e e a d e GIC a e.

Note

I e e e e , a d e GICD_CPENDSGIR e e , a SGI de f ed b e c b a f SGI be a d ce ce .

Usage constraints A e e b c e d a e e ed SGI RAZ/WI.

T e e e e a e b e-acce b e.

If e GIC e e e Sec E e :

a e e b a c e d a G 0 e RAZ/WI N - ec e acce e

f e GIC fe e a e ce , e e b c e d e - e e ed ce a e RAZ/WI.

Note

I a ce e e a , e ce acce e e e ca c a e e SGI e d a e c e d e face. C a e e d a f a SGI f e a e ce d e affec e a f a SGI a e ce .

PPI a d SPI b e e I e Se -Pe d e e , GICD_ISPENDR .

Configurations T e e e e a e e e GIC 2. T e e e ca a e e e ed GIC 1.

F SGI Se -Pe d e e a e e e ed. T e e e c a ab f eac f e b e ce ce , f eac f e 16 b e SGI . T a , eac e e c a e e - e d b f eac f f SGI .

I a ce e e a , e GICD_SPENDSGIR e e a e ba ed f eac c ec ed ce .

Attributes See e e e a Tab e 4-1 a e 4-75.

F e 4-19 e GICD_SPENDSGIR b a e .

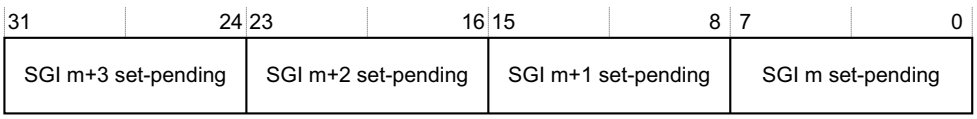


Figure 4-19 GICD_SPENDSGIR bit assignments

Tab e 4-24 e GICD_SPENDSGIR b a e .

Table 4-24 GICD_SPENDSGIRn bit assignments

Bits	Name	Function		
[8 +7:8], f =0 3	SGI Se - e d b	F eac b :		
		Reads	**0**	SGI f e c e d ce e d [a].
			1	SGI f e c e d ce e d [a].
		Writes	**0**	Ha effec .
			1	Add e e d a e f SGI f e c e d ce , f a ead e d . If SGI a ead e d f e c e d ce e e e a effec .
		See e f e e a be ee e SGI be , , e GICD_SPENDSGIR e e be , , a d e f e d be , .		
		Note A acce e e a e SGI a a e e ce a e acce .		

a. Pe d e c de e a a e ac e a d e d .

F SGI ID , e e a ed b CPU *C* GICD_SGIR, e DIV a d MOD a e e e e d a d d e a :

e c e d GICD_SPENDSGIR e e be , , e b = DIV 4

e ff e f e e ed GICD_SPENDSGIR (0xF20 + (4*))

e SGI Se - e d f e d ff e , , e b = MOD 4

e e ed b e SGI Se - e d f e d b C.

4.3.18 Identification registers

T a c ec e ec f ca def e ff e 0xFD0-0xFFC e D b e e a a a ead-de f ca e e ace. Tab e 4-25

Tab e 4-26 e ICPIDR2 b a e .

Table 4-26 ICPIDR2 bit assignments

Bits	Name	Function
[31:8]	-	IMPLEMENTATION DEFINED. T e C eL a d C eS Pe e a ID Re e c e e e e e e b be e e ed, RAZ, a d ARM ec e d a e e a f c e e.
[7:4]	A c Re	Re f e d f e GIC a c ec e. T e a e f f e d de e d e GIC a c ec e e : 0x1 f GIC 1 0x2 f GIC 2.
[3:0]	-	IMPLEMENTATION DEFINED.

The ARM implementation of the GIC Identification Registers

Note

T e ARM e e a f e e e e c e e de f ca c e e f C eL a d C eS c e . T e e a de f e e de ce a a GIC a e e a c ec e. I d e de f e de e a fac e f e GIC e e a . F f a ab e de e a d a fac e f a GIC e e a ee e GICD_IIDR a d GICC_IIDR de c .

I e c e , de f ca c e e de f e a c e a e . T e GIC e f e c e e d ffe e . I de f e a e de ce a e e a fa e f e GIC a c ec e def ed b ec f ca . S f a e ead e GICD_IIDR a d GICC_IIDR d c e, f e a e, e e e e a d e f e GIC a d a e.

Tab e 4-27 e Ide f ca Re e f a ARM e e a f e e f e GIC a c ec e def ed b ec f ca . ARM ec e d e e e e c de e e e de c bed e e.

Table 4-27 Identification Registers for a GIC, with ARM implementation values

Register[a]	Offset	Bits	ARM implementation Value	Description
C e ID0, ICCIDR0	0xFF0	[7:0]	0x0D	ARM-def ed f ed a e f e ea b e f c e d c e .
C e ID1, ICCIDR1	0xFF4	[7:0]	0xF0	
C e ID2, ICCIDR2	0xFF8	[7:0]	0x05	
C e ID3, ICCIDR3	0xFFC	[7:0]	0xB1	
Pe e a ID0, ICPIDR0	0xFE0	[7:0]	0x90	B [7:0] f e ARM-def ed De ID f e d.
Pe e a ID1, ICPIDR1	0xFE4	[7:4]	0xB	B [3:0] f e ARM-def ed A c ID f e d.
		[3:0]	E a e a e : 0x3 f ARM GIC 1 e e a 0x4 f ARM GIC 2 e e a .	B [11:8] f e ARM-def ed De ID f e d:

Table 4-27 Identification Registers for a GIC, with ARM implementation values (continued)

Register[a]	Offset	Bits	ARM implementation Value	Description
Pe e a ID2, ICPIDR2	0xFE8	[7:4]	A c ec a -def ed: 0x1 f GIC 1 0x2 f GIC 2.	A c Re f e d.
		[3]	1	ARM-def ed U e JEPc de f e d.
		[2:0]	0b011	B [6:4] f e ARM-def ed A c ID f e d.
Pe e a ID3, ICPIDR3	0xFEC	[3:0]	0x0	Re e ed b ARM.
		[7:4]	0x0	ARM-def ed Re f e d.
Pe e a ID4, ICPIDR4	0xFD0	[3:0]	0x4	ARM-def ed C a C de f e d.
		[7:4]	0x0	Re e ed b ARM.
Pe e a ID5, ICPIDR5	0xFD4	[7:0]	0x00	Re e ed b ARM.

Table 4-27 Identification Registers for a GIC, with ARM implementation values (continued)

Register[a]	Offset	Bits	ARM implementation	
			Value	Description
Pe e a ID6, ICPIDR6	0xFD8	[7:0]	0x00	Re e ed b ARM.
Pe e a ID7, ICPIDR7	0xFDC	[7:0]	0x00	Re e ed b ARM.

a. I e ARM e e a , b [31:8] f eac e e a e e ed. B [7:0] f e f C e ID e e e e def e a c ce a 32-b C e ID, a d b [7:0] f e e Pe e a ID e e e e def e ac ce a 64-b Pe e a ID. I e GIC e e a , de e e a e, C e ID a d Pe e a ID efe e a c ec e f e e e a , ee e N e a e a f ec f e f a .

Note

S e e ARM e e a f e GIC d d e e Pe e a ID e e 4-7. S f a e ca e e a e f b [3] f e ICPIDR2 de f e e e e a :

0 Le ac f a .

1 ARM GIC 1 a e f a .

The ARM peripheral ID for a GIC

T e e, e Pe e a ID e e ICPIDR0 ICPIDR7 def e a 64-b e e a ID. I c e ARM e e a , b [63:36] f a ID a e e e ed, RAZ. F e 4-21 b [35:0] f e Pe e a ID f a GIC, a d Tab e 4-28 a e f e d e 64-b Pe e a ID.

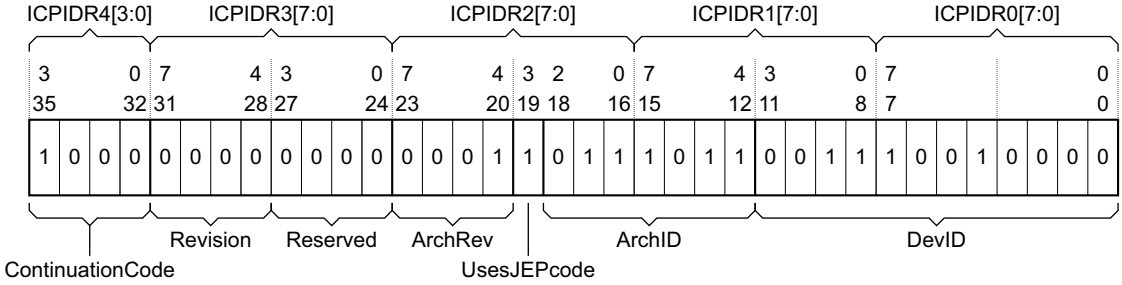


Figure 4-21 ARM Peripheral ID fields for a GIC

Table 4-28 Fields in the GIC Peripheral ID, for an ARM implementation

Name	Bits	Source	Function, ARM-defined fields
-	[39:36]	ICPIDR4[7:4]	Re e ed
C a C de	[35:32]	ICPIDR4[3:0]	JEP106 c a c de f ARM
Re	[31:28]	ICPIDR3[7:4]	Re f e d
-	[27:24]	ICPIDR3[3:0]	Re e ed
A c Re	[23:20]	ICPIDR2[7:4]	A c ec a -def ed e be f e ARM GIC a c ec e, ee *P ID2 R , ICPIDR2* a e 4-119

Table 4-28 Fields in the GIC Peripheral ID, for an ARM implementation (continued)

Name	Bits	Source	Function, ARM-defined fields
U e JEPc de	[19]	ICPIDR2[3]	I d ca e a e de f e e JEP106 c de de f ARM a e de e f e a c ec e
A c ID	[18:12]	ICPIDR2[2:0], ICPIDR1[7:4]	Ide f e ARM a e de e f e GIC a c ec e
De ID	[11:0]	ICPIDR1[3:0], ICPIDR0[7:0]	Ide f e e de ce a a a c a GIC e e a

4.4 CPU interface register descriptions

T e f ec de c be e CPU e face e e :

See *CP* a e 4-76 f add e ff e a d e e f a f e e e e .

4.4.1 CPU Interface Control Register, GICC_CTLR

T e GICC_CTLR c a ac e c a e:

Purpose E ab e e a f e b e CPU e face e c ec ed ce , a d de add a - e e c f e CPU e face. I a GIC 2 e e a , c de c f e (EOI) be a .

Note

I a GIC 2 e e a a c de e GIC Sec E e , de e de EOI c a e ded f :

Acce e f Sec e a e. T c a e e a d f b G 0 a d G 1 e .

Acce e f N - ec e a e. T c a e e a d f G 1 e .

T e EOI c affec e be a f acce e GICC_EOIR a d GICC_DIR. See e e e de c f e f a .

Usage constraints If e GIC e e e Sec E e f c f a c d , e e ca e e e acce ce a e e f e d e Sec e GICC_CTLR, ee *C* a e 4-82.

Configurations If e e e a e , e e de de e de c f G 0 a d G 1 e .

If e GIC e e e Sec E e :

e e ba ed de Sec e a d N - ec ec e , ee *R* a e 4-77

e e e b a e a e d ffe e e Sec e a d N - ec ec e f e e e , a d:

G e Sec e c f e e e ca c b G 0 a d G 1 e

G e N - ec e c f e e e ca c G 1 e .

Attributes See e e e a Tab e 4-2 a e 4-76.

F e 4-22 a e 4-126 a d Tab e 4-29 a e 4-126 e GICC_CTLR b a e f a GIC 1 e e a , f a e e a a d e c de e Sec E e e N - ec ec f e e e, a e e a a c de e Sec E e .

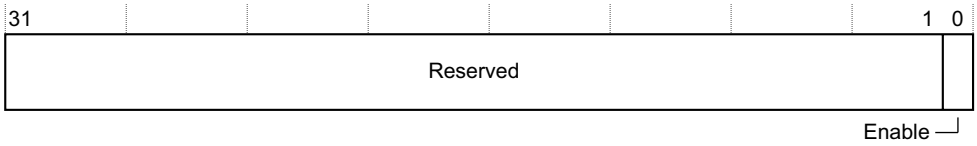


Figure 4-22 GICC_CTLR bit assignments, GICv1 without Security Extensions or Non-secure

Table 4-29 GICC_CTLR bit assignments, GIC1 without Security Extensions or Non-secure

Bits	Name	Function
[31:1]	-	Re e ed
[0]	E ab e	E ab e f e a f G 1 e b e CPU e face e c ec ed ce . **0** D ab e a f e **1** E ab e a f e . **Note** W e b c ea ed 0, e CPU e face e a e d e f a ded . W e b e 1, e CPU e face a ce e d e a a e f a ded . T e e a a b f e e e ed f a c a e a e effec . O a GIC 1 e e a a d e c de e Sec E e , b c e a f a e b e CPU e face e c ec ed ce . See *E D CP* a e 4-77 f e f a ab b .

F e 4-23 a d Tab e 4-30 e GICC_CTLR b a e f e N - ec e c f e e e a GIC 2 e e a a c de e Sec E e ,

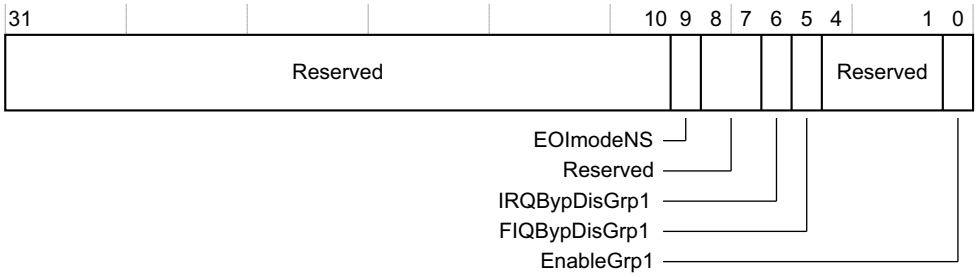


Figure 4-23 GICC_CTLR bit assignments, GICv2 with Security Extensions, Non-secure copy

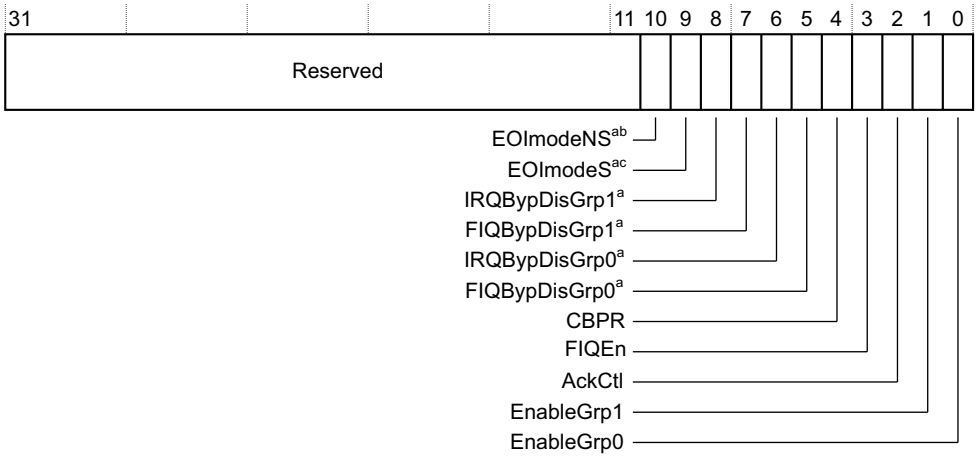
Table 4-30 GICC_CTLR bit assignments, GIC2 with Security Extensions, Non-secure copy

Bits	Name	Function
[31:10]	-	Re e ed
[9]	EOI deNS	C e be a f N - ec e acce e e GICC_EOIR a d GICC_DIR e e : **0** GICC_EOIR a b d a d deac a e e f c a . Acce e e GICC_DIR a e UNPREDICTABLE. **1** GICC_EOIR a d f c a . T e GICC_DIR e e a deac a e e f c a . See *B GICC_EOIR, GIC 2* a e 4-140 f e f a .
[8:7]	-	Re e ed.

Table 4-30 GICC_CTLR bit assignments, GIC2 with Security Extensions, Non-secure copy (continued)

Bits	Name	Function
[6]	IRQB D G 1	W e e a f IRQ b e CPU e face d ab ed, b a c e e e b a IRQ a a ed e ce : **0** B a IRQ a a ed e ce **1** B a IRQ a a ed e ce . See *I , GIC 2* a e 2-27 f e f a .
[5]	FIQB D G 1	W e e a f FIQ b e CPU e face d ab ed, b a c e e e b a FIQ a a ed e ce : **0** B a FIQ a a ed e ce **1** B a FIQ a a ed e ce . See *I , GIC 2* a e 2-27 f e f a .
[4:1]	-	Re e ed
[0]	E ab eG 1	E ab e f e a f G 1 e b e CPU e face e c ec ed ce . **0** D ab e a f e **1** E ab e a f e . **Note** W e b e 0, e CPU e face e a e d G 1 e f a ded . W e b e 1, e CPU e face a ce e d G 1 e a a e f a ded . T e e a a b f e e e ed f ac a e a e effec . See *E D CP* a e 4-77 f e f a ab b .

F e 4-24 a e 4-128 a d Tab e 4-31 a e 4-128 e GICC_CTLR b a e f :
a GIC 2 e e a , f :
G a e e a a d e c de e Sec E e
G e Sec e c f e e e , a e e a a c de e Sec E e
a GIC 1 e e a a c de e Sec E e , f e Sec e c f e e e .



a GICv2 only
b When the GIC implementation includes the Security Extensions
c EOImode in a GIC implementation that does not include the Security Extensions

Figure 4-24 GICC_CTLR bit assignments, GICv2 without Security Extensions or Secure

Table 4-31 GICC_CTLR bit assignments, GICv2 without Security Extensions or Secure

Bit	Name	Function
[31:11]	-	Re e ed.
[10]	EOI deNS[ab]	A a f EOI deNS f e N - ec e c f e e , ee Tab e 4-30 a e 4-126. I a GIC 2 e e a a d e c de e Sec E e , a d a GIC 1 e e a , b e e ed.
[9]	EOI deS[a]	C e be a f acce e GICC_EOIR a d GICC_DIR e e . I a GIC e e a a c de e Sec E e , c a e Sec e acce e , a d e EOI deNS b c e be a f N - ec e acce e e e e : **0** GICC_EOIR a b d a d deac a e e f c a . Acce e e GICC_DIR a e UNPREDICTABLE. **1** GICC_EOIR a d f c a . GICC_DIR a deac a e e f c a . See *B GICC_EOIR, GIC 2* a e 4-140 f e f a . **Note** T b ca ed EOI de a GIC e e a a d e c de e Sec E e . I a GIC 1 e e a , b e e ed.
[8]	IRQB D G 1[a]	A a f IRQB D G 1 f e N - ec e c f e e , ee Tab e 4-30 a e 4-126. I a GIC 1 e e a , b e e ed.
[7]	FIQB D G 1[a]	A a f FIQB D G 1 f e N - ec e c f e e , ee Tab e 4-30 a e 4-126. I a GIC 1 e e a , b e e ed.

Table 4-31 GICC_CTLR bit assignments, GICv2 without Security Extensions or Secure (continued)

Bit	Name	Function
[6]	IRQB D G 0[a]	W e e a f IRQ b e CPU e face d ab ed, b a c e e e b a IRQ a a ed e ce : **0** B a IRQ a a ed e ce **1** B a IRQ a a ed e ce . See *I , GIC 2* a e 2-27 a d *P , GIC 2* a e 2-31 f e f a . I a GIC 1 e e a , b e e ed.
[5]	FIQB D G 0[a]	W e e a f FIQ b e CPU e face d ab ed, b a c e e e b a FIQ a a ed e ce : **0** B a FIQ a a ed e ce **1** B a FIQ a a ed e ce . See *I , GIC 2* a e 2-27 a d *P , GIC 2* a e 2-31 f e f a . I a GIC 1 e e a , b e e ed.
[4]	CBPR[c]	C e e e GICC_BPR de c c G 0 a d G 1 e . **0** T de e e a ee , e: e GICC_BPR f G 0 e e GICC_ABPR f G 1 e . **1** T de e e a ee e e GICC_BPR f b G 0 a d G 1 e . See *T* a e 3-57 f e f a ab GICC_CTLR.CBPR affec acce e GICC_BPR a d GICC_ABPR.
[3]	FIQE	C e e e CPU e face a G 0 e a a e ce e FIQ e IRQ a . **0** S a G 0 e e IRQ a . **1** S a G 0 e e FIQ a . T e GIC a a a G 1 e e IRQ a . **Note** If f a e e f a e a c de a e e a f GIC 1 e Sec E e , a e a e a ed b e IRQ a . I c e , e e a b 0. T e defa a e. See *E GIC* a e 3-68 f e f a .

Table 4-31 GICC_CTLR bit assignments, GICv2 without Security Extensions or Secure (continued)

Bit	Name	Function
[2]	Ac C	W e e e d e a G 1 e , de e e b : e e a ead f GICC_IAR ac ed e e e , e a e ID e e a ead f GICC_HPPIR e e ID f e e d e , e a e ID. **0** If e e d e a G 1 e , a ead f e GICC_IAR e GICC_HPPIR e a I e ID f 1022. A ead f e GICC_IAR d e ac ed e e e , a d a effec e e d a f e e . **1** If e e d e a G 1 e , a ead f e GICC_IAR e GICC_HPPIR e e I e ID f e G 1 e . A ead f GICC_IAR ac ed e a d Ac a e e e . I a GIC e e a a c de e Sec E e , c affec e be a f Sec e e e acce e . F e f a , ee: *T* a e 3-50 *I* a e 3-53 *B* *GICC EOIR*, *GIC 1 S E* a e 4-139 *E* *S E* *GICC HPPIR* a e 4-143. **Note** ARM de eca e e f GICC_CTLR.Ac C , a d ec e d a f a e de e e GICC_CTLR.Ac C e 0. See *E D CP* a e 4-77 f e f a ab e effec f e b .
[1]	E ab eG 1[d, e]	E ab e f e a f G 1 e b e CPU e face e c ec ed ce : **0** D ab e a f G 1 e . **1** E ab e a f G 1 e .
[0]	E ab eG 0[e, f]	E ab e f e a f G 0 e b e CPU e face e c ec ed ce : **0** D ab e a f G 0 e . **1** E ab e a f G 0 e .

a. GIC 2 , ee *P* , *GIC 2* a e 2-31 f e f a .
b. W e e GIC e e e Sec E e .
c. SBPR GIC 1.
d. E ab eNS GIC 1.
e. T e e a a b f e e e ed f ac a e e a e f e e a e effec .
f. E ab eS GIC 1.

F e f a ab e a f e a b a , c d e GIC 2 d ab e c
f f c a , ee *I* , *GIC 2* a e 2-27.

4.4.2 Interrupt Priority Mask Register, GICC_PMR

T e GICC_PMR c a ac e c a e:

Purpose P de a e f e. O e e a e a e e e a e a ed e ce .

Note

H e c e d a e P f e d a e.

Usage constraints If e GIC e e e Sec E e e :

a N - ec e acce e e ca ead e a a e a c e d e e a f f e a e, ee *I* a e 3-53.

f a Sec e e a a ed e GICC_PMR a a e a c e d a a e e e a f f e a e e :

G a N - ec e ead f e GICC_PMR e `0x00`, e a d e f e a e e d e e e

G a N - ec e e e GICC_PMR ed.

F e f a ee *N - G 0* a e 4-81.

W e de e e ee , e a e ca be a , e B a P e e , GICC_BPR.

Configurations T e e a a ab e a c f a f e GIC. If e GIC e e e Sec E e , e e C .

Attributes See e e e a Tab e 4-2 a e 4-76.

F e 4-25 e GICC_PMR b a e .

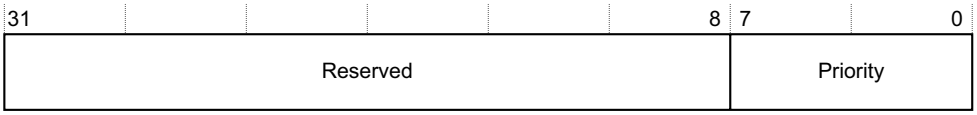


Figure 4-25 GICC_PMR bit assignments

Tab e 4-32 e GICC_PMR Re e b a e .

Table 4-32 GICC_PMR Register bit assignments

Bits	Name	Function
[31:8]	-	Re e ed.
[7:0]	P	T e a e e f e CPU e face. If e f a e e a e a e d ca ed b f e d, e e face a e e e ce . If e GIC fe e a 256 e e e e b a e RAZ/WI, a f : **128 supported levels** B [0] = 0. **64 supported levels** B [1:0] = `0b00`. **32 supported levels** B [2:0] = `0b000`. **16 supported levels** B [3:0] = `0b0000`. F e f a ee *I* a e 3-44.

T e f e d c de e effec f e GIC Sec E e acce e e e :

```
// MaskRegRead()
// =============

bits(8) MaskRegRead()

    read_value = GICC_PMR<7:0>;
    if NS_access then                                   // A non-secure GIC access.
        if read_value<7> == '0' then
            read_value = '00000000';                    // A secure priority value, RAZ
        else
            read_value = LSL((read_value AND P_MASK), 1);
    return(read_value);


// MaskRegWrite()
// ==============

MaskRegWrite(bits(8) value)

    if NS_access then                                   // A non-secure GIC access.
        mod_write_val = (‘10000000’ OR LSR(value,1)) AND P_MASK;
        if GICC_PMR<7> == '1' then                      // Non-secure execution can only update the
            GICC_PMR[cpu_id]<7:0> = mod_write_val;      // Priority Mask Register if the current
                                                        // value is in the range 0x80 to 0xFF
        else
            IgnoreWriteRequest();
    else                                                // A secure GIC access
        GICC_PMR<7:0> = value AND P_MASK;
```

4.4.3 Binary Point Register, GICC_BPR

T e GICC_BPR c a ac e c a e:

Purpose T e e def e e a c e a e f e d a , e f e d a d e f e d. T e f e d ed de e e e ee . F e f a ee *P* a e 3-45 a d *P* a e 3-45.

Usage constraints T e b a a e IMPLEMENTATION DEFINED e a e:

0-3 f e e e a d e c de e GIC Sec E e , a d f e Sec e c f e e e f e e e a c de e Sec E e

1-4 f e N - ec e c f e e e .

A a e a eb a f e d a a e e a e a e e f e d e a e. O a e e, eb a f e d e e ed a e.

Configurations T e e a a ab e a c f a f e GIC. If e GIC e e e Sec E e :

e e ba ed de Sec e a d N - ec e c e , ee *R* a e 4-77

e GICC_ABPR a a a f e N - ec e c f GICC_BPR

e GICC_CTLR.CBPR b affec e e f e N - ec e GICC_BPR.

I a GIC 2 e e a , a GIC 1 e e a a c de e Sec E e , GICC_CTLR.CBPR c e e e Sec e c f e GICC_BPR, e GICC_ABPR, ed f e ee f G 1 e .

See *P* a e 3-45 a d *T* a e 3-57 f e f a .

Attributes See e e e a Tab e 4-2 a e 4-76.

F e 4-26 e GICC_BPR b a e .

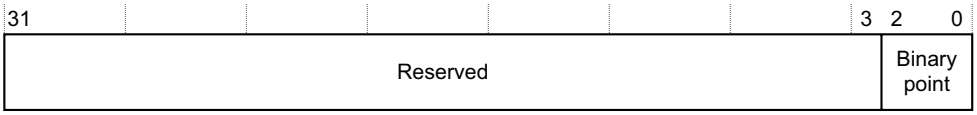


Figure 4-26 GICC_BPR bit assignments

Tab e 4-33 e GICC_BPR b a e .

Table 4-33 GICC_BPR bit assignments

Bits	Name	Function
[31:3]	-	Re e ed.
[2:0]	B a	T e a e f f e d c e 8-b e f e d a f e d, ed de e e e ee , a d a b f e d. F f e d d e e e e e b a ed e f e d ee: Tab e 3-7 a e 3-57, f e ce f G 1 e a GIC a e , e e GICC_CTLR.CBPR b e 1 Tab e 3-2 a e 3-46, f a e ca e . See *P* a e 3-45 f e f a .

Note

A a e N - ec e GICC_BPR a e GICC_ABPR ea a , a ce e , a ce a ca a e Sec e acce e e GIC ca acce e GICC_ABPR, c f e e ee e f G 1 e .

T e f e d c de e effec f e GIC Sec E e acce e e e :

```
// BinaryPointRegWrite()
// =====================

BinaryPointRegWrite(bits(3) value)

    if NS_access && GICC_CTLR.CBPR == '1' then
        IgnoreWriteRequest();
    else
        GICC_BPR<2:0> = value;                          // Banked register

bits(3) BinaryPointRegRead()

    read_value = GICC_BPR<2:0>;                         // Banked register
    if NS_access && GICC_CTLR.CBPR == '1' then
        read_value = GICC_BPR_Secure;                   // The secure copy of the BPR
        if read_value != 7 then
            read_value = read_value + 1;
    return(read_value);
```

4.4.4 Interrupt Acknowledge Register, GICC_IAR

T e GICC_IAR c a ac e c a e:

Purpose T e ce ead e e b a e e ID f e a ed e . T ead ac a a ac ed e f e e .

Usage constraints W e GICC_CTLR.Ac C e 0 a GIC 2 e e a a d e c de e Sec E e , f e e e d e G 1, e e ID 1022 e ed.

Configurations T e e a a ab e a c f a f e GIC. If e GIC e e e Sec E e :
e e C .
e GICC_AIAR a a a f e N - ec e e f e e .

Attributes See e e e a Tab e 4-2 a e 4-76.

F e 4-27 e IAR b a e .

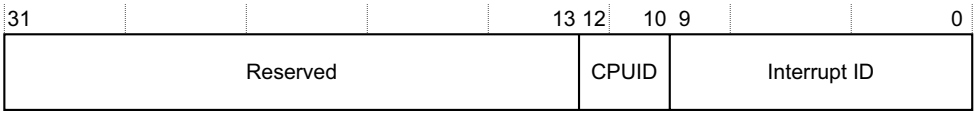


Figure 4-27 GICC_IAR bit assignments

Tab e 4-34 e IAR b a e .

Table 4-34 GICC_IAR bit assignments

Bit	Name	Function
[31:13]	-	Re e ed.
[12:10]	CPUID	F SGI a ce e e a , f e d de f e e ce a e e ed e e . I e e be f e CPU e face a ade e e e , f e a e a a e f 3 ea e e e a e e a e d b a e e GICD_SGIR CPU e face 3. F a e e f e d RAZ.
[9:0]	I e ID	T e e ID.

A ead f e GICC_IAR e e e ID f e e e d e f e CPU e face. T e ead e a e ID f 1023 f a f e f a :
f a d f e b e D b e CPU e face d ab e d
a f e b e CPU e face e c ec ed ce d ab e d
e d e e CPU e face a ff c e f e e face a e ce .

Note

T e f e e ce f e e a e a e f e e GIC e a e ID f 1023, a d ead f e GICC_IAR ca be c ca :

1. A e e a a e a e e - e e e .
2. T e e a ff c e a d e ef e e GIC a a a e d ce .
3. T e e e a dea e e e e . Beca e e e e e d e f ff c e , e GIC dea e e e e e e ce .

4. Bef e a ec ed e dea e f e e e f a e 3, e a e ed ce ead e GICC_IAR. Beca e e e ff c e a e ce , e GIC e e ID a e f 1023.

T e de e a f e e ed e ID e c e f e GIC e , ee *E GICC IAR*.

A - e ID e ed b a ead f e GICC_IAR ca ed a a d e ID.

W e e GIC e a a d e ID a ead f e GICC_IAR ea e ead a a ac ed e f a e a d, a a de-effec f e ead, c a e e e a f e d ac e, ac e a d e d f e e d a e f e e . N a , e e d a e fa e e f e e e e - e ea d e a a e ed.

F e e ead fa a d I e ID f e GICC_IAR, e c ec ed ce e f a a c e e GICC_EOIR.

Note

F c a b b e e e e GIC a c ec e ec f ca , ARM ec e d a f a e e e e e e e e a e ead f e GICC_IAR, a d e a a e bac e GICC_EOIR e a c e ed ce f e e .

A e a e ce a e ead e GICC_IAR a a e, GIC 2 e ce ca b a a a d e ID, ee *I I-N* a e 3-41 f e f a .

Effect of interrupt grouping on reads of the GICC_IAR

Note

T ec d e a GICV_IAR, e c e d e e e a CPU e face.

W e a GIC e e a e , e e a ead f e GICC_IAR e a a d e ID de e d :

e e e e a e d e f ff c e f be a ed e ce , a d f , e e :

G e e e d e a G 0 a G 1 e

G e a e ab ed f a e .

f e GIC e e e Sec E e , e e e GICC_IAR ead acce Sec e N - ec e e a e f e GICC_CTLR.Ac C b .

Read f e GICC_IAR a d e a a d e ID e a e ID, ID 1022 1023, ee *S GIC* a e 3-50. Tab e 4-35 a b e GICC_IAR ead f a GIC a e a CPU e face a e e e Sec E e . F a GIC 2 CPU e face a d e e e e Sec E e , a e e e ce e f N - ec e GICC_IAR ead a .

Table 4-35 Effect of interrupt grouping and the Security Extensions on reads of GICC_IAR

State	GICC_IAR read	GICC_CTLR.AckCtl	Returned interrupt ID
H e e d e [a] G 1	N - ec e		ID f G 1 e
	Sec e	1	ID f G 1 e
		0	I e ID 1022

Table 4-35 Effect of interrupt grouping and the Security Extensions on reads of GICC_IAR (continued)

State	GICC_IAR read	GICC_CTLR.AckCtl	Returned interrupt ID
H e e d e [a] G 0	N - ec e		I e ID 1023
	Sec e		ID f G 0 e
N e d e [a]			I e ID 1023
I e a f e e ed e b CPU e face d ab ed			I e ID 1023

a. Of ff c e be a ed e ce f a b e CPU e face e ab ed.

T e f e d c de e effec f e GIC Sec E e acce e e :

```
// ReadGICC_IAR()
// ==============
//
// Value of GICC_IAR read by a CPU access
//

bits(32) ReadGICC_IAR(integer cpu_id)

    pendID = HighestPriorityPendingInterrupt(cpu_id);

    if ( IsGrp0Int(pendID) && (GICD_CTLR.EnableGrp0 == '0' || GICC_CTLR.EnableGrp0 == '0')) ||
       (!IsGrp0Int(pendID) && (GICD_CTLR.EnableGrp1 == '0' || GICC_CTLR.EnableGrp1 == '0'))
    then
        pendID = 1023;                  // If the highest priority isn't enabled, then no interrupt

    if pendID != 1023 then              // An enabled interrupt is pending
        if IsGrp0Int(pendID) then       // Highest priority is Group 0
            if NS_access then
                pendID = 1023;
        else                            // Highest priority is Group 1
            if !NS_access && (GICC_CTLR[cpu_id].AckCtl == '0') then
                pendID = 1022;

    cpuID = 0;                          // Must be zero for non-SGI interrupts

    if pendID < 16 then                 // 0 .. 15 are Software Generated Interrupts
        sgiID = SGI_CpuID(pendID);      // value is IMPLEMENTATION DEFINED

    if pendID < 1020 then               // Check that it is not a spurious interrupt
    AcknowledgeInterrupt(pendID);       // Set active and attempt to clear pending

    rval = 0;
    rval<12:10> = sgiID;
    rval<9:0> = pendID;

    return(rval);
```

4.4.5 End of Interrupt Register, GICC_EOIR

T e GICC_EOIR c a ac e c a e:

Purpose A ce e e f e CPU e face e e :

a a c e ed e ce f e ec f ed e

a GIC 2 e e a , e e a a e GICC_CTLR.EOI de b e 1, d ca e a e e face d e f d f e ec f ed e .

See *P* a e 3-38 f e f a .

Usage constraints A e e e c e d e ece a d ead f a I e Ac ed e Re e . A a d ead a ead a e a a d e ID, a a e ID.

Configurations T e e a a ab e a c f a f e GIC. If e GIC e e e Sec E e e e C .

Attributes See Tab e 4-2 a e 4-76.

F e 4-28 e GICC_EOIR b a e .

Figure 4-28 GICC_EOIR bit assignments

Tab e 4-36 e GICC_EOIR b a e .

See *P* a e 3-38 f e f a ab e effec f a e e e .

F e e ead f a a d I e ID f e GICC_IAR, e c ec ed ce e f a a c e e GICC_EOIR. T e a e e e GICC_EOIR be e e ID ead f e GICC_IAR.

If a ead f e GICC_IAR e e ID f a e , f a e d e a e a e a c e d

F e ed e , e de f e GICC_EOIR be e e e f e de f e ac ed e e . Be a UNPREDICTABLE f e e :

e de c a ead f e GICC_IAR a d e e GICC_EOIR a e a a ed e a e a e e GICC_EOIR d e a c a ac e e , e ID f a e .

T e effec f GICC_EOIR a a d e ID UNPREDICTABLE f a f e f a :

e a e e d e a c e a a d e a e ead f e I e Ac ed e e e e a d ac ed ed e e d ca ed e a a ead bee b ec a EOI e e .

F a e e a e a a GIC 1 e e a e GIC Sec E e , ee e f e f ec f e f a :

B GICC EOIR, GIC 1 S E .

B GICC EOIR, GIC 2 a e 4-140.

Behavior of writes to GICC_EOIR, GICv1 with Security Extensions

If a CPU e face a GIC 1 e e a e e e GIC Sec E e , e e a e e GICC_EOIR e e e ac e a f e de f ed e de e d :

e e e de f ed e G 0 G 1

e e e GICC_EOIR e Sec e N - ec e

e a e f e GICC_CTLR.Ac C b .

Tab e 4-37 a b e e f a e e GICC_EOIR.

Table 4-37 Effect of the Security Extensions on writes to GICC_EOIR

Interrupt status	GICC_EOIR write	GICC_CTLR.AckCtl	Active status removed
G 0	N - ec e		N
	Sec e		Ye
G 1	N - ec e		Ye
	Sec e	1	Ye
	Sec e	0	UNPREDICTABLE

W e GICC_CTLR.Ac C == 0, e de e e e f GICC_EOIR e e a e GICC_IAR ead a e de e de f Sec e a d N - ec e e e acce e . T ea :

a Sec e e GICC_EOIR c e d e ece Sec e ead f GICC_IAR

a N - ec e e GICC_EOIR c e d e ece N - ec e ead f GICC_IAR

a Sec e e e GICC_AEOIR c e d e ece Sec e ead f e GICC_AIAR.

W e GICC_CTLR.Ac C == 1, e de e e e f Sec e GICC_EOIR e e a e GICC_IAR ead a e acc f e ec e e f e GICC_IAR acce e . T ea a a Sec e e GICC_EOIR c e d e ece ead f GICC_IAR, e a d e f e ec e e f a ead f GICC_IAR.

Note

T e a e f GICC_CTLR.Ac C a effec e be a f N - ec e e e acce e . A N - ec e e GICC_EOIR c e d e ece N - ec e ead f GICC_IAR. H e e , e GICC_CTLR.AcKC e 0, N - ec e f a e e f a GICC_IAR e f a e f Sec e f a e a a ead e f ed e GICC_IAR e f a e . If d e , e effec f e e UNPREDICTABLE.

Behavior of writes to GICC_EOIR, GICv2

See *P* a e 3-38 f e e a f a ab e effec f e GICC_EOIR, a d ab e b e e a a f e GIC d a d e deac a e e a a GIC 2 e e a .

I a GIC 2 e e a , e GICC_CTLR.Ac C e 0:

GICC_EOIR ed f ce G 0 e
GICC_AEOIR ed f ce G 1 e .

I a GIC 2 e e a a c de e GIC Sec E e :

GICC_CTLR.EOI deS c e be a f Sec e acce e GICC_EOIR a d GICC_AEOIR

GICC_CTLR.EOI deNS c e be a f N - ec e acce e GICC_EOIR

e GICC_CTLR.Ac C e 0:

G a N - ec e e GICC_EOIR c e d e ece N - ec e ead f GICC_IAR

G a Sec e e e GICC_AEOIR c e d e ece Sec e ead f e GICC_AIAR.

Tab e 4-38 , f a GIC 2 e e a , e ec e e f e GICC_EOR acce , a d e a e f e GICC_CTLR.Ac C b , de e e e P d effec f a a d GICC_EOIR e. I a , a e a e e ed e e a f e Ac e P e Re e , e d c ea ab e e e Sec e Ac e P e Re e , GICC_APR , N - ec e Ac e P e Re e , GICC_NSAPR . If e GIC d e e e e GIC Sec E e , e e e f e Sec e GICC_EOIR acce e a .

Table 4-38 Priority drop effect of GICC_EOIR writes

GICC_EOIR access	GICC_CTLR.AckCtl	Highest priority active interrupt	Effect
N - ec e	-	G 1	Pe f d f G 1 e . I e Ac e P e e e , c ea e e ac e G 1 e e
N - ec e	-	G 0	A c ec a UNPREDICTABLE. T acce affec e e f ac e G 0 e e . **Note** T e e a e a IMPLEMENTATION DEFINED effec . F e a e, a e e a c ea e e ac e G 1 e e e Ac e P e e e .
Sec e	0	-	Pe f d f G 0 e . I e Ac e P e e e , c ea e e ac e G 0 e e.
Sec e	1	-	Pe f d . T e , a d d , a e acc f e . I e Ac e P e e e , c ea e e ac e e e . T ca be e e a G 0 a G 1 ac e de e d c e e . If e e ac e e e f b G 0 a d G 1 a e e a e, e effec UNDEFINED.

Tab e 4-39 , f a GIC 2 e e a , e ec e e f e GICC_EOR acce , a d e a e f e GICC_CTLR c b , de e e e a a d GICC_EOIR e deac a e e de f ed e . If e GIC d e e e e GIC Sec E e , e e f e N - ec e GICC_EOIR acce e d a .

Table 4-39 Deactivate interrupt effect of GICC_EOIR writes

GICC_EOIR access	GICC_CTLR.AckCtl	EOImode bit[a]	Identified interrupt	Effect
N - ec e	-	0	G 1	I e deac a ed
N - ec e	-	0	G 0	Acce ed
Sec e	-	0	G 0	I e deac a ed
Sec e	1	0	G 1	I e deac a ed
Sec e	0	0	G 1	UNPREDICTABLE
-	-	1	-	I e e a ac e

a. F a GIC 2 e e a a d e c de e Sec E e .

F a GIC 2 e e a a c de e Sec E e , GICC_CTLR.EOI de ca ed:

GICC_CTLR.EOI deS f Sec e acce e GICC_EOIR. T e a a e Sec e acce e GICC_AEOIR

GICC_CTLR.EOI deNS f N - ec e acce e GICC_EOIR.

4.4.6 Running Priority Register, GICC_RPR

T e GICC_RPR c a ac e c a e:

Purpose I d ca e e R f e CPU e face.

Usage constraints If e e ac e e e CPU e face, e a e e ed e Id e .

Note

S f a e ca de e e e be f e e ed b f a ead f e e .

If e GIC e e e Sec E e , e a e e ed b a N - ec e ead f e P f e d :

0x00 f e f e d a e e a 0x80

e N - ec e e f e P a e f e f e d a e 0x80 e.

F e f a ee *N - G 0* a e 4-81.

Configurations T e e a a ab e a c f a f e GIC. If e GIC e e e Sec E e e e C .

Attributes See e e e a Tab e 4-2 a e 4-76.

F e 4-29 e GICC_RPR b a e .

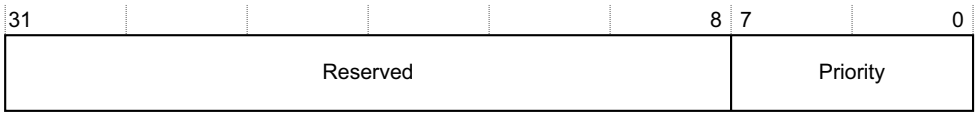


Figure 4-29 GICC_RPR bit assignments

Tab e 4-40 e GICC_RPR b a e .

Table 4-40 GICC_RPR bit assignments

Bit	Name	Description
[31:8]	-	Re e ed.
[7:0]	P	T e c e e CPU e face.

T e f e d c de e effec f e GIC Sec E e acce e e e :

```
// ReadGICC_RPR()
// ==============
//
// Value of GICC_RPR read by a processor access
//

bits(8) ReadGICC_RPR()

    read_value = GICC_RPR<7:0>;
    if NS_access then                                   // A non-secure GIC access,
        if read_value<7> == '0' then                    //     therefore, adjust value.
            read_value = '00000000';                    // A secure priority value, RAZ
        else
            read_value = LSL((read_value AND P_MASK), 1);
    return(read_value);
```

4.4.7 Highest Priority Pending Interrupt Register, GICC_HPPIR

T e GICC_HPPIR c a ac e c a e:

Purpose I d ca e e I e ID, a d ce ID f a a e, f e e e d e e CPU e face.

Usage constraints Ne e e e I e ID f a e a ac ea d e d . Re a ce ID f a SGI a ce e e a .

If e GIC e , e a e e ed b a ead f GICC_HPPIR ca de e d :

e a e f GICC_CTLR.Ac C

f e GIC e e e Sec E e , e e e e e acce Sec e N - ec e:

See *E S E GICC HPPIR*.

Configurations T e e a a ab e a c f a f e GIC. If e GIC e e e Sec E e e e C .

Attributes See e e e a Tab e 4-2 a e 4-76.

F e 4-30 e GICC_HPPIR b a e .

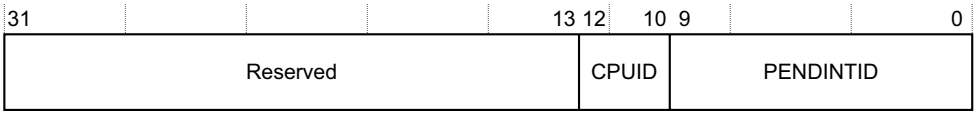


Figure 4-30 GICC_HPPIR bit assignments

Tab e 4-41 e GICC_HPPIR b a e .

Table 4-41 GICC_HPPIR bit assignments

Bit	Name	Description
[31:13]	-	Re e ed.
[12:10]	CPUID	O a ce e e a , f e PENDINTID f e d e e ID f a SGI, f e d c a e CPUID a e f a e . T de f e e ce a e e a ed e e . I a e ca e f e d RAZ.
[9:0]	PENDINTID	T e e ID f e e e d e . See Tab e 4-42 a e 4-144 f e f a ab e e f N - ec e ead f e GICC_HPPIR e e GIC e e e Sec E e .

Effect of interrupt grouping and the Security Extensions on reads of the GICC_HPPIR

If a CPU e face e , e e a ead f e GICC_HPPIR e a a d e ID de e d :

e e e e e d e c f ed a a G 0 a G 1 e e a e f e GICC_CTLR.Ac C b .

f e GIC e e e Sec E e , e e e GICC_HPPIR ead acce Sec e N - ec e.

Read f e GICC_HPPIR a d e a a d e ID e a e ID, ID 1022 1023, ee *S GIC* a e 3-50. Tab e 4-42 a b e GICC_HPPIR ead f a CPU e face a e e e Sec E e a GIC a e . If e CPU e face d e e e e Sec E e , e e e a a Sec e GICC_HPPIR ead de c be e be a .

Table 4-42 Effect of the Security Extensions on GICC_HPPIR reads

Current state	GICC_HPPIR read	GICC_CTLR.AckCtl	Returned interrupt ID
H e e d e G 1	N - ec e		ID f G 1 e
	Sec e	0	S e ID 1022
		1	ID f G 1 e
H e e d e G 0	N - ec e		S e ID 1023
	Sec e		ID f G 0 e
N e d e			S e ID 1023

T e f e d c de e effec f e GIC Sec E e acce e e e :

```
// ReadGICC_HPPIR()
// ================
//
// Value of GICC_HPPIR read by a CPU access

bits(32) ReadGICC_HPPIR(integer cpu_id)
    // cpu_id identifies the accessed CPU interface
    // GICC_CTLR[cpu_id] is the GICC_CTLR register for that interface

    pendID = HighestPriorityPendingInterrupt(cpu_id);

    if ( IsGrp0Int(pendID) && GICD_CTLR.EnableGrp0 == '0') ||
       (!IsGrp0Int(pendID) && GICD_CTLR.EnableGrp1 == '0')
    then
        pendID = 1023;                  // If required group is not enabled, then no interrupt

    if GICC_MASK_HPPIR                  // GICC_MASK_HPPIR indicates the IMPLEMENTATION DEFINED
                                        // choice whether GICC_CTLR.EnableGrp{0,1} being zero
    then                                // returns a spurious interrupt
        if ( IsGrp0Int(pendID) && GICC_CTLR[cpu_id].EnableGrp0 == '0') ||
           (!IsGrp0Int(pendID) && GICC_CTLR[cpu_id].EnableGrp1 == '0')
        then
            pendID = 1023;              // If required group is not enabled, then no interrupt

    if pendID != 1023 then              // An enabled interrupt is pending
        if IsGrp0Int(pendID) then       // Highest priority is Group 0
            if NS_access then
                pendID = 1023;
        else                            // Highest priority is Group 1
            if !NS_access && (GICC_CTLR[cpu_id].AckCtl == '0') then
                pendID = 1022;

    cpuID = 0;                          // Must be zero for non-SGI interrupts

    if pendID < 16 then                 // 0 .. 15 are Software Generated Interrupts
        sgiID = SGI_CpuID(pendID);      // value is IMPLEMENTATION DEFINED

    rval = 0;
    rval<12:10> = sgiID;
    rval<9:0> = pendID;

    return(rval);
```

4.4.8 Aliased Binary Point Register, GICC_ABPR

T e GICC_ABPR c a ac e c a e:

Purpose A B a P Re e f a d G 1 e .

T e e e a e f e e def ed a (GICC_BPR.B a + 1), e a e ed a e f 0x1-0x4.

Usage constraints If e GIC e e e Sec E e , acce b e b Sec e acce e .

Configurations T e e e e GIC 2, a d GIC 1 e e a a c de e Sec E e ,

I a GIC e e a a c de e Sec E e , GICC_ABPR a a a f e N - ec e GICC_BPR, a d e GICC_CTLR.CBPR e 0, a Sec e acce e e e a e a N - ec e acce GICC_IAR.

Note

GICC_ABPR ed da e GICC_CTLR.CBPR e 1. I a GIC e e a a c de e Sec E e , e GICC_CTLR.CBPR e 1, e be a f Sec e acce e GICC_ABPR de ca e be a f N - ec e acce e GICC_BPR

Acce e e GICC_ABPR a e affec ed b e a e f e GICC_CTLR.CBPR b .

If e GIC e e a c de e Sec E e , GICC_ABPR a Sec e e e. If e GIC d e e e e GICC_ABPR, e add e RAZ/WI.

Attributes See e e e a Tab e 4-2 a e 4-76.

F e 4-31 e GICC_ABPR b a e .

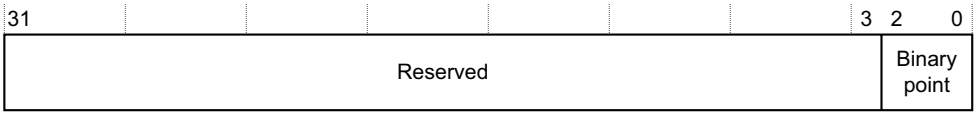


Figure 4-31 GICC_ABPR bit assignments

Tab e 4-43 e GICC_ABPR b a e .

Table 4-43 GICC_ABPR bit assignments

Bits	Name	Function
[31:3]	-	Re e ed.
[2:0]	B a	T e a e f f e d c e 8-b e f e d a f e d, ed de e e e ee , a d a b f e d. F f e d de e e e e b a ed e f e d ee: Tab e 3-7 a e 3-57, f e ce f G 1 e a GIC a e , e e GICC_CTLR.CBPR b e 1 Tab e 3-2 a e 3-46, f a e ca e . See *P* a e 3-45 f e f a .

Note

I a GIC e e a a c de e Sec E e , a a e N - ec e GICC_BPR a e GICC_ABPR ea a , a ce e , a ce a ca a e Sec e acce e e GIC ca acce e GICC_ABPR, c f e e ee e f G 1 e .

4.4.9 Aliased Interrupt Acknowledge Register, GICC_AIAR

T e GICC_AIAR c a ac e c a e:

Purpose A I e Ac ed e e e f a d G 1 e .

T e ce ead e e b a e e ID f e a ed G 1 e . T ead ac a a ac ed e f e e .

Usage constraints If e GIC e e e Sec E e , acce b e b Sec e acce e .

Configurations T e e e e GIC 2. If e GIC e e e Sec E e , GICC_AIAR a a a f e N - ec e e f GICC_IAR, a d a Sec e acce e e de ca a N - ec e acce GICC_IAR.

If e GIC e e e Sec E e a Sec e e e .

Attributes See e e e a Tab e 4-2 a e 4-76.

F e 4-32 e GICC_AIAR b a e .

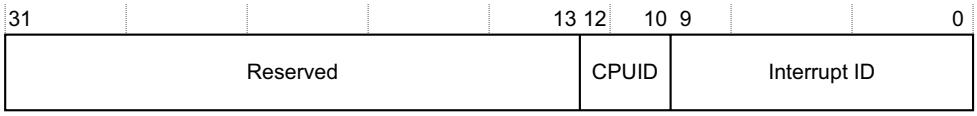


Figure 4-32 GICC_AIAR bit assignments

Tab e 4-44 e GICC_AIAR b a e .

Table 4-44 GICC_AIAR bit assignments

Bit	Name	Function
[31:13]	-	Re e ed, SBZ.
[12:10]	CPUID	F SGI a ce e e a , f e d de f e e ce a e e ed e e . I e e be f e CPU e face a ade e e e , f e a e a a e f 3 ea e e e a e e a ed b a e e GICD_SGIR CPU e face 3. F a e e f e d RAZ.
[9:0]	I e ID	T e e ID.

4.4.10 Aliased End of Interrupt Register, GICC_AEOIR

T e GICC_AEOIR c a ac e c a e:

Purpose A e d f e e e f a d G 1 e .

W e GICC_CTLR.Ac C e 0, a e e e e f d f e de f ed G 1 e , a d f e a a e GICC_CTLR EOI de b e 0, a deac a e e e e . F e f a ee *P* a e 3-38.

Note

I a GIC a e e e , e GICC_CTLR.Ac C e 0 e Sec e GICC_EOIR ca be ed f e beca e a e a e e affec GICC_APR , GICC_NSAPR .

Usage constraints A e e e c e d e ece ac ed ed G 1 e .

If e GIC e e a c de e Sec E e , a Sec e e e .

Configurations T e e e e GIC 2.

If e GIC e e e Sec E e , GICC_AEOIR effec e a a a f e N - ec e GICC_EOIR. A Sec e acce e e a a N - ec e acce GICC_EOIR, e ce a e GICC_CTLR.EOI deS b ed. See *E I R , GICC EOIR* a e 4-138 f e f a .

If e GIC e e e Sec E e , a Sec e e e .

Attributes See e e e a Tab e 4-2 a e 4-76.

F e 4-33 e GICC_AEOIR b a e

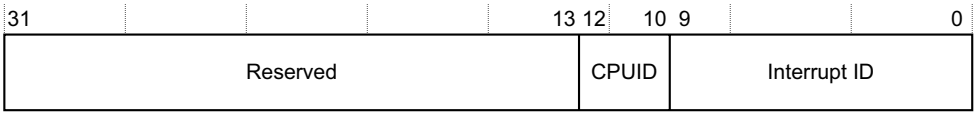


Figure 4-33 GICC_AEOIR bit assignments

Tab e 4-45 e GICC_AEOIR b a e .

Table 4-45 GICC_AEOIR bit assignments

Bit	Name	Function
[31:13]	-	Re e ed, SBZ.
[12:10]	CPUID	O a ce e e a , e ce a SGI, f e d c a e CPUID a e f e c e d GICC_AIAR, N - ec e GICC_IAR, acce . I a e ca e f e d SBZ.
[9:0]	I e ID	T e I e ID a e f e c e d GICC_AIAR, N - ec e GICC_IAR, acce .

T e effec UNPREDICTABLE f a a e e a e a a e ead f e GICC_AIAR e GICC_AEOIR.

4.4.11 Aliased Highest Priority Pending Interrupt Register, GICC_AHPPIR

T e GICC_AHPPIR c a ac e c a e:

Purpose P de a H e P Pe d I e e e f e a d f G 1 e .
If e e e d e e CPU e face a G 1 e , e e e ID f a e . O e e, e a e ID f 1023.

Usage constraints Ne e e e I e ID f a e a ac e a d e d . If e GIC e e e Sec E e , acce b e b Sec e acce e .

Configurations T e e e e GIC 2.
If e GIC e e e Sec E e , GICC_AHPPIR a a a f e N - ec e GICC_HPPIR, a d a Sec e acce e e e a e a N - ec e acce GICC_HPPIR.
If e GIC e e e Sec E e a Sec e e e .

Attributes See e e e a Tab e 4-2 a e 4-76.

F e 4-34 e GICC_AHPPIR b a e .

Figure 4-34 GICC_AHPPIR bit assignments

Tab e 4-46 e GICC_AHPPIR b a e .

Table 4-46 GICC_AHPPIR bit assignments

Bit	Name	Description
[31:13]	-	Re e ed.
[12:10]	CPUID	O a ce e e a , f e PENDINTID f e d e e ID f a SGI, f e d c a e CPUID a e f a e . T de f e e ce a e e a ed e e . I a e ca e f e d RAZ.
[9:0]	PENDINTID	T e e ID f e e e d e , f a e a G 1 e . O e e, e e ID, 1023.

4.4.12 Active Priorities Registers, GICC_APRn

T e GICC_APR c a ac e c a e:

Purpose T e ea e IMPLEMENTATION DEFINED e e a de f e e a d e e ac e e- a a e e e e a .

Usage constraints A e f a f e e e e IMPLEMENTATION DEFINED:

beca e GIC 2 a a ee e ab a e a d e e a GIC a e, e GICC_APR e e be e e a GIC e e a

a e e a a c de e GIC Sec E e , N - ec e acce e affec Sec e e a , a d e a c ec e e e a e e e a e ba ed, de Sec e a d N - ec ec e f e e e .

Configurations T e e e a e e e GIC 2. T e e e ca a e e e ed GIC 1.

T e be f Ac e P e e e e e ed de e d e be f P ee e e ed, ee Tab e 4-47 a e 4-150. If e GIC d e e e e Sec E e , e e e e d e ac e e f e G 0 e .

Note

T e GICC_NSAPR e e e a a d e ac e e f e G 1 e .

If e GIC e e e Sec E e , e e e e a e ba ed de Sec e a d N - ec ec e , ee *R* a e 4-77. T e e a:

N - ec e acce e d affec Sec e e a

e N - ec ec e f e e e e de a N - ec e e f e e f e G 1 e , ee *S GIC S E* a e 3-53.

e Sec ec e f e e e e ac ac e e f G 0 e .

Note

T e Sec ec e f e GICC_NSAPR e e ac ac e e f G 1 e .

Attributes See e e e a Tab e 4-2 a e 4-76.

Note

T e e e e a e IMPLEMENTATION DEFINED, b ARM ec e d a , e e e e e a e e e ed, e f e de e ec .

If e GIC e e a c de e Sec E e , e f e Ac e P e Re e ace b e N - ec e acce e a a e c e e G 1 e e e e a , ee *S GIC S E* a e 3-53 a d *T* a e 3-57. T ea e e a f f e P ee e e ace b e, b e a ed a ea e e a f f e ee e e ace, a Tab e 4-47 a e 4-150 .

See *P* a e 3-45 f e f a ab a d ee e e .

P ed c ab e e e a a a eed f e GICC_APR e e a e a ed bef e e d a d e ed af e e . H e e, f ad ffe e a e e e e e e, fa e a e e e e e d e a , be a UNPREDICTABLE.

T e e ab :

ARM ec e d a f a e e e e e f e e f a a d e a e

b e GICC_APR a d e GICC_NSAPR c de a b f eac ee e e a e e e .

I a e e a a c de e GIC Sec E e :

N -Sec e e e acce e acce b c e d ee e e e N -Sec e ace, b e e N - ec e e f e e , ee *S GIC S E* a e 3-53

e GICC_NSAPR e e de e D b e f e ac e e f e G 1 e .

Tab e 4-47 e GICC_APR e e a .

Table 4-47 Active Priorities register implementation

Minimum value of Secure GICC_BPR	Minimum value of Non-secure GICC_BPR	Maximum number of group priority bits	Maximum number of preemption levels	GICC_APRn implementation	View of Active Priorities Registers for Non-secure accesses[a]
3	4	4	16	GICC_APR0[15:0]	GICC_NSAPR0[15:8] a ea a GICC_APR0[7:0]
2	3	5	32	GICC_APR0[31:0]	GICC_NSAPR0[31:16] a ea a GICC_APR0[15:0]
1	2	6	64	GICC_APR0-GICC_APR1	GICC_NSAPR1 a ea a GICC_APR0
0	1	7	128	GICC_APR0-GICC_APR3	GICC_NSAPR2-GICC_NSAPR3 a ea a GICC_APR0-GICC_APR1

a. I a GIC e e a a c de e GIC Sec E e .

4.4.13 Non-secure Active Priorities Registers, GICC_NSAPRn

T e GICC_NSAPR c a ac e c a e:

Purpose T e ea e IMPLEMENTATION DEFINED e e a de f e e a d e eac e e- a a e e e a . T e eae e a a e e f G 1 e .

S f a e ec ec ed ce ca a e e e ec e eac e e a e b :

e GICC_APR e e a e e e e a ef e G 0 e

e GICC_NSAPR e e a e e e e a ef e G 1 e .

I a e e a a c de e Sec E e :

e e e e e a N - ec e acce e ca e fe e Sec e e a

Sec e f a e ec ec ed ce ca a e e e ec e eac e e a e e GICC_APR a d e GICC_NSAPR e e .

Usage constraints A e f a f e e e e IMPLEMENTATION DEFINED:

beca e GIC 2 a a ee e ab a e a d e a GIC a e, e GICC_NSAPR e e be e e a GIC e e a

a e e a a c de e Sec E e , e e e e a e acce b e b Sec e acce e .

Configurations T e e e e a e e e e GIC 2 e e a a c de e GIC Sec E e . T e e e ca a e e e ed GIC 1.

T e ea e Sec e e e .

T e be f Ac e P e e e e e ed de e d e be f e e f ee e e ed, ee Tab e 4-47 a e 4-150.

Attributes See e e e a Tab e 4-2 a e 4-76.

T a e e Sec e f a e e e a e G 1 e e G 0 ace, e e e e be a e e c a e e e ed ace. T e ef e, e e e ed GICC_NSAPR e e e de ca f a e GICC_APR e e e, b c a e ac e e f G 1 e a e a G 0 e . If e GIC e e a c de e Sec E e , e e a f f e e e e ca be acce ed b N - ec e f a e acce e N - ec ec e f e GICC_APR e e , b e e acce e e a N - ec e e f e e e , a Tab e 4-47 a e 4-150 .

See *A P R , GICC APR* a e 4-149 e f a ab e e e a f e e e e .

4.4.14 CPU Interface Identification Register, GICC_IIDR

T e GICC_IIDR c a ac e c a e:

Purpose P de f a ab e e e e a d e f e CPU e face.

Usage constraints N a e c a .

Configurations T e e a a ab e a c f a f e GIC. If e GIC e e e Sec E e e e C .

Attributes See e e e a Tab e 4-2 a e 4-76.

F e 4-35 e GICC_IIDR b a e .

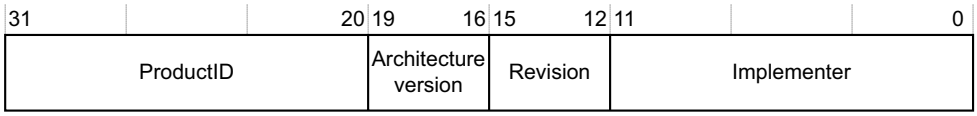


Figure 4-35 GICC_IIDR bit assignments

Tab e 4-48 e GICC_IIDR b a e .

Table 4-48 GICC_IIDR bit assignments

Bit	Name	Description
[31:20]	P d c ID	A IMPLEMENTATION DEFINED d c de f e .
[19:16]	A c ec e e	T e a e f f e d de e d e GIC a c ec e e , a f : `0x1` f GIC 1 `0x2` f GIC 2.
[15:12]	Re	A IMPLEMENTATION DEFINED e be f e CPU e face.
[11:0]	I e e e	C a e JEP106 c de f e c a a e e ed e GIC CPU e face:[a] **Bits [11:8]** T e JEP106 c a c de f e e e e . **Bit [7]** A a 0. **Bits [6:0]** T e JEP106 de c de f e e e e .

a. F a ARM e e a , e a e f f e d `0x43B`.

4.4.15 Deactivate Interrupt Register, GICC_DIR

T e GICC_DIR c a ac e c a e:

Purpose W e e d e a a ed f e deac a , a de c bed *P* a e 3-38, a e e deac a e e ec f ed e .

Usage constraints W e e e a e a effec e :

f a GIC e e a a d e c de e Sec E e , GICC_CTLR.EOI de e 1

f a GIC e e a a c de e Sec E e :

G f Sec e acce e e e e , GICC_CTLR.EOI deS e 1

G f N - ec e acce e e e e , GICC_CTLR.EOI deNS e 1.

If e e e a EOI de b 0 e e effec f e e acce UNPREDICTABLE.

If e e de f ed e GICC_DIR ac e, a d a e , e effec f e e e e UNPREDICTABLE. T ea a GICC_DIR e de f a e f c e e a bee a a d GICC_EOIR GICC_AEOIR e.

U e GICC_EOIR a d GICC_AEOIR e , e e de e e e f GICC_DIR e , ded e ee e e e e e e ec .

ARM ec e d a e a e e GICC_DIR e 32-b a e e ed b e c e d GICC_IAR GICC_AIAR ead.

Configurations T e e e e GIC 2. T e e e ca e e ed GIC 1. If e GIC e e e Sec E e e e C .

Attributes See e e e a Tab e 4-2 a e 4-76.

Note

T e e e e e deac a e e a a c a de .

F e 4-36 e GICC_DIR b a e .

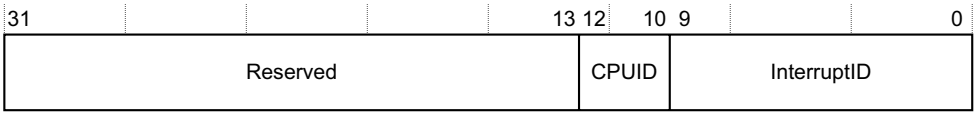


Figure 4-36 GICC_DIR bit assignments

Tab e 4-49 e GICC_DIR b a e .

Table 4-49 GICC_DIR bit assignments

Bit	Name	Description
[31:13]	-	Re e ed, SBZ
[12:10]	CPUID	F a SGI a ce e e a , f e d de f e e ce a e e ed e e . F a e e f e d RAZ.
[9:0]	I e ID	T e e ID

Behavior of writes to the GICC_DIR

Re a d e f e a e f e GICC_CTLR.Ac C b :

f e e e a d e c de e Sec E e , a a d e GICC_DIR deac a e e ec f ed e , e a d e f e e a e G 0 G 1

f e e e a c de e Sec E e , a a d:

- Sec e e GICC_DIR deac a e e ec f ed e , e a d e f e e a e G 0 G 1
- N - ec e e GICC_DIR deac a e e ec f ed e f a e G 1. A a d e e a ec f e a e a ac e, a d f c e e a bee a cce f e GICC_EOIR GICC_AEOIR.

Tab e 4-50 e be a f a d e GICC_DIR. I a e e a a d e c de e Sec E e , a d e a e e be a f Sec e GICC_DIR e .

Table 4-50 Behavior of GICC_DIR writes

GICC_CTLR.AckCtl	GICC_DIR write	Interrupt group	Effect
	N - ec e	G 1	I e deac a ed.
	N - ec e	G 0	W e ed.
	Sec e		I e deac a ed.

4.5 Preserving and restoring GIC state

S e a e e e e a e a d e e e a e f e GIC, a c a e . F e a e: d e d
e a e a e f e CPU a e a ce e e a
a e a ce a a , e c a be ee a ac e .

T f c a , GIC 2 de :

T e ab a e a d e e e ac e a e e d GICD_ISACTIVER a d GICD_ICACTIVER .

Note

I e e a c a add a a e, c a ac e b e ce ce f SGI , beca e c add a a e e ed d e e e a . F e a e, e a a e e ea d e e GIC a e b a e Se -ac e b f GICD_ISACTIVER be ca ab e f a a d e a ac e a e f a .

T e ab e a d c ea e d SGI , ded b GICD_CPENDSGIR a d GICD_SPENDSGIR .

Ac e P e e e , GICC_APR a d GICC_NSAPR , a de e ab acce d ec e a c a ed GIC a c ec e a e, a d a e a d e e a a e.

GIC Support for Virtualization

T c a e de c be e GIC V a a E e d ced GIC 2, a d e e e a GIC a e a ea e ce a e e e ARM V a a E e . I c a e f ec :

5.1 About implementing a GIC in a system with processor virtualization

W a GIC a c de e GIC V a a E e , a a ac e a ce a c de e ARM 7-A V a a E e c ca e a a CPU e face e GIC. T e a ac e ece e a e f e face, a d ca d e e e f ca e .

A e a d e a IRQ , a a e de ed f a a ac e a e , a d, c c e GIC, a a e e a e a d e a c a ed ca e . I a e e GIC a e face c e e a a e e a CPU e face. A a f c , e e da e e L e e , a a e a b e f e GIC a e face c e e . I a e e a d GIC e e de a a d b , a a ea a a ac e a e ca GIC d b .

T e GIC a CPU e face a a e e a ac e, b ec e a GIC a d a d a e . F e 5-1 a e 5-159 a e a e f e GIC a d e e a e e a a ce a a .

Note

A ARM ce e e a a c de e V a a E e a c de e Sec E e . S c a ce a e e ed a GIC a e e b e GIC Sec E e a d GIC V a a E e . T e e a e c a e de c be c a e e a , f c :

- G G 0 ca e a e Sec e e
- G G 1 ca e a e N - ec e e .
- G e e e f e a ce f a ca IRQ , a e a e ed a a IRQ a FIQ
- G Sec e M de e f e a ce f a ca FIQ .

See *S E* a e 1-16 f e f a .

I de c f ce a a , a a ac e a G e OS, a a ca . I a c e , e e a d *G OS* a e .

I e ARM de f a N - ec e e a f a ce a e e e ARM V a a E e , Sec e f a e e ce c f e e e a de c bed *IRQ FIQ N - S* a e 3-68, a FIQ a e ed f Sec e e , a d IRQ f N - ec e e .

W e e e ece e a IRQ, de e e e e e e f e f, f a a ac e. If f a a ac e de e e c a ac e a de e e a d e e a e a a e , ee *M GIC CP* a e 5-160.

T e GIC V a a E e de e f f a a CPU e face:

GIC a e face c e e . T e e a e a a e e e e , acce ed b a e , a f a e. See *M GIC CP* a e 5-160 f e f a .

GIC a CPU e face e e . T e e e e de e a CPU e face acce ed b e c e a ac e a c ec ed ce . I e e a , e a e e a e f a a e GIC ca CPU e face e e , b e e a e e e e def ed b e L e e .

A a ac e c ca e e a CPU e face, b ca de ec a c ca a GIC ca CPU e face.

T e a CPU e face a d e GIC a e face c e e a e b e N - ec e e a . A e e e N - ec e a e 2 add e a a e e a e a ac e ca acce e GIC a e face c e e . T , e GIC a c ec e e e e GIC a CPU e face e e a d e GIC a e face c e e be e a a e 4KB add e e . See e *ARM A R M* , *ARM 7-A ARM 7-R* f e f a ab e ARM V a a E e a d N - ec e add e a a .

GIC a e 4-74 de c be e a c e f e GIC e e , c ea a :
a a e a GICH_ d ca e a e e de c bed *GIC* a e 5-167
a a e a GICV_ d ca e a e e de c bed *GIC* *CP* a e 5-179.

A a ca CPU e face, e a CPU e face a a e f e e e d e a ff c e . H e e, e a ed e a a e a e a a ca e . T e L e e d ca e e e e e G 0 G 1 a d e ef e e e a ed a a a IRQ a a FIQ.

T e a ac e:
ac ed e a a e b ead f e I e Ac ed e Re e .
d ca e e a c e ed e ce b e E d f I e Re e .

T e e e da e e L e e . See *A* a e 5-162 f e f a .

F e 5-1 e de f e e a GIC a ARM 7-A ce a e e e ce V a a E e .

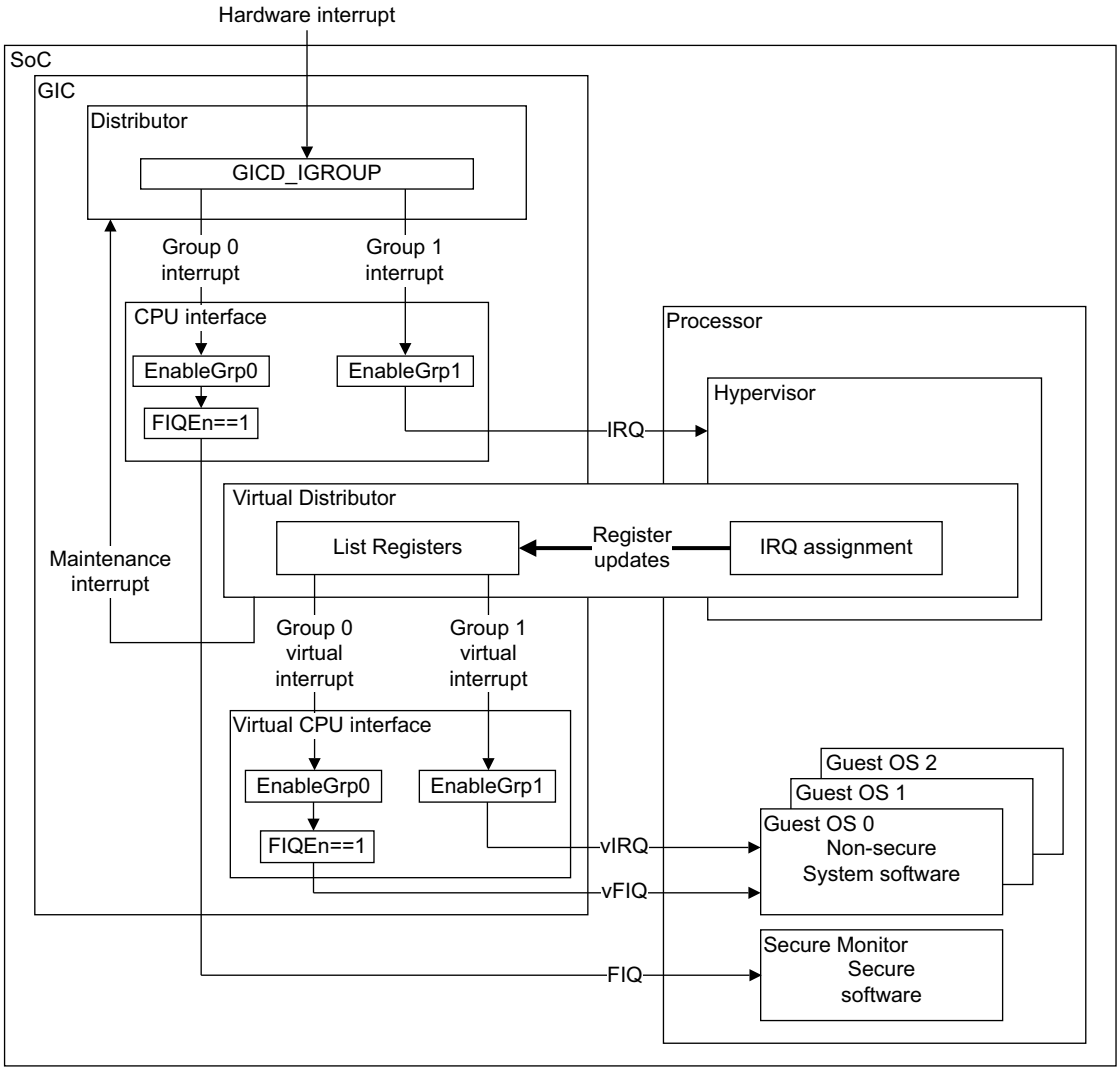


Figure 5-1 Implementing the GIC with an ARM processor that supports virtualization

5.2 Managing the GIC virtual CPU interface

T ec de c be e ARM de f a ARM 7-A ce a e e e ce V a a E e .

T e e , a f a e, a a e e GIC a e face c e e , c f:

List registers U ed def e e ac e a d e d a e f e a CPU e face. T e c e a ac e acce e e e e d ec , e a CPU e face.

Management registers

U ed a a e e a CPU e face, a d a e a d e e e e c be ee a ac e .

O e ce , e e :

c f e IRQ be a e H de, a a d e a IRQ e f

e e a e 2 N - ec e add e a a :

G a a G e OS acce e e GIC D b e e , a ca de e e e a d b e f eac a ac e

G e e a e a ac e ca acce e GIC a e face c e e

G e a e GIC CPU e face e e add e ace e GIC a CPU e face e e .

c f e e e ed a e a ce e f e a CPU e face, ee *M* a e 5-164.

T e e c , a d c e be ee , e a ac e . W e a a a ac e, a e L e e e def e e e a a e b e a a ac e.

W e ece e a ca IRQ, e e de e e e e ed de a f e e a d e e e :

P ce e e e e f, f e a e f e IRQ a a e a ce e f e a CPU e face. I e deac a e e ca e .

Ge e a e a a e . De e d e e a d e a e ed a ac e, e e a e e f e f ac :

G If e e f ec e a ac e, da e e L e e de a f e e , edef e e a a e b e ec e a ac e. If e e ace e L e e , a e ec e e e de a ca be added a a a e a e. See *L* a e 5-161 f e f a .

G Rec d a e e f ad ffe e a ac e b a de a f e e a a f e e a e a c a ed a a ac e.

G S c e ad ffe e a ac e a ca a de e e e . I d a e e e a e f ec e a ac e, e f a e L e e , a d e a e L e e e , d ca e e e a e f e e a ac e, c d e a e f e e a a a ed.

T e a ac e acce e e GIC a CPU e face e e . T e e e e a e e a a e e e a f a a e ca CPU e face e e , a d, a ca e e a e a ac e be e e acce a ca CPU e face. T e e acce e da e e a e a d a b e L e e .

W e e a ac e a d e a a e , e e a CPU e face d ca e e a f ed ce . T e a CPU e face a c e e ca D b a d e ca D b e deac a e e e .

Note

T e e a f e GIC a c ec e. I ed b e ARM 7-A A c ec e V a a E e . T e e a N - ec e f a e H de. See e *ARM A R M* , *ARM 7-A* *ARM 7-R* f e f a .

T e e a de f ec de c be :

L
C
A a e 5-162
GIC a e 5-164
M a e 5-164
S - a e 5-165
GIC *E* a e 5-165.

5.2.1 List registers and virtual interrupt handling

W a GIC e e a a c de e V a a E e , a e e L e e a a e f e a e . T e a be f e a a e e e d , ac e, ac e a d e d , ca e ceed e be f L e e a a ab e. If a e , e e ca a e e e ac e e e e e , a d a e e e e e L e e , ba ed e . T e ef e:

T e L e e c de a ac e, ac e a d e d , e . V a CPU e face acce e b e a ac e da e e L e e , a d a a EOI e e f e a ac e deac a e a e e . H e e, e a ac e ca e a EOI e e f a e bef e e e e e e a ca e d ac e e e a L e e . I ca e, e EOI e e ca da e e L e e .

Note

O e - e e a e d e ca be ac e a d e d .

A e L e e c de ac e e , e e a a a e d e e e , a e d e ca be a e d e a ac e e e add e L e e e . T e ef e, e e a e c a de e eff c e a ac e e a , ARM ec e d a e L e e c a a a ea e e d e , ded a L e e a a ab e f e .

T a a e 1-N e a d de , a e a e a e a e f e a ac e a e . A e a e f e be ece a e e ab d ab a a CPU e face.

T e GIC V a a E e c de e f fea e a e a d :

P d f c a e a a e f e deac a , ee *C* .

Ma e a ce e a e e e , ee *M* a e 5-164.

5.2.2 Completion of virtualized physical interrupts

T ec de c be e ced e a ARM ec e d f e c e f a ca e a a bee a ed. U e e ced e ed ce e a f e e e e ed, ea e e ca ece e e ca e a ea a b e. T e e ca e f be e a e e a ed e e .

W e a ca e , ARM ec e d a , f eac CPU e face a c e d a ce a ac e , e GICC_CTLR.EOI deNS b e 1. T ea a e acce e e GICC_EOIR e e d e f e CPU e face b d e deac a e e e . Af e e EOI e e , e e e e CPU e face e , a b e e e ca be a e d e ce .

ARM ec e d a ca e c e c f e f e a a e e :

1. EOI
2. e deac a .

T e e e a e e a ed e de a a f :

1. Af e ece a ca e , e e e f a EOI e e f e ca e b e GICC_EOIR GICC_AEOIR e e . Af e EOI, a e a ac e a ce ed e a e , e e f e CPU e face ea a e e ca ece e e ca e .

   T e ab e e e e e f e e a e a e, d e f e b a e EOI ced e f - a ed ca e de c bed *G* a e 3-37.

   Note

   T e e a a e a ed e e a e e ca e ece b ec EOI. T beca e e e a e bee deac a ed. T e e e e f be e- a ed e e bef e be ce ed b e a ac e.

2. Af e e a ac e c e e ce e c e d a e , e e GICV_EOIR GICV_AEOIR deac a e e e . T deac a e b e a e a d e c e d ca e , ded a b f e f c d a e e:

   - e GICV_CTLR.EOI de b e 0
   - e GICH_LR .HW b e 1.

   A e a e , f e GICV_CTLR.EOI de b e 1, e a ac e e e GICV_DIR e e deac a e e e .

   If e GICH_LR .HW b e 0, e e deac a e e ca e e f. ARM ec e d e f e f e d f deac a ca SGI a a e ed a a ac e:

   - e e deac a e e SGI b e GICC_DIR e e af e e a ac e e GICC_EOIR
   - e e e a EOI a e a ce e e e GICC_DIR e e af e e a ac e e GICV_EOIR, ee *M* a e 5-164 f e f a .

T e e a f e e e e :

GICV_IAR and GICV_AIAR

T e a ac e ead GICV_IAR GICV_AIAR ac ed e a e . A e ID e ed e :

e e e ac ed e

a e e ead be ac ed ed e e .

T e a GIC e a e c e e e de a e a e ac ed ed a CPU e face a e b e ca a d a CPU e face .

GICV_EOIR and GICV_AEOIR

T e EOI e e e e I e ID a d CPUID a e ead e e e a ac ed ed. A e e a a e e e , GICV_EOIR GICV_AEOIR c ea e ee b a c a ed e e ac e e e Ac e P e Re e , GICH_APR:

W e e e ac e e a G 0 e , e a a e a e ead f GICV_IAR GICV_EOIR:

G c ea e ee b GICH_APR

G f GICV_CTLR.EOI de c ea ed 0, e e e ac e a e e c e d L e e

G f GICV_CTLR.EOI de c ea ed 0 a d e GICH_LR .HW b e 1, deac a e e c e d ca e e D b

W e e e ac e e a G 0 e , e effec f GICV_AEOIR UNPREDICTABLE.

W e e e ac e e a G 1 e , e a a e a e ead f GICV_AIAR GICV_AEOIR:

G c ea e ee b GICH_APR, a d

G f GICV_CTLR.EOI de c ea ed 0, e e e ac e a e e c e d L e e

G f GICV_CTLR.EOI de c ea ed 0 a d e GICH_LR .HW b ==1, deac a e e c e d ca e e D b

W e e e ac e e a G 1 e , e effec f GICV_EOIR UNPREDICTABLE.

Tab e 4-37 a e 4-139 GICV_AEOIR affec ed b GICV_CTLR.Ac C .

GICV_HPPIR and GICV_AHPPIR

F e a CPU e face:

a ead f GICV_HPPIR e e G 0 e d e e e

a ead f GICV_AHPPIR e e G 1 e d e e e .

Tab e 4-42 a e 4-144 GICV_HPPIR affec ed b GICV_CTLR.Ac C .

T e e e e GICC_CTLR.EOI de b e a a e d e ca CPU e face a d e deac a e D b . T e e ca e GICC_DIR deac a e e , e e e f e D b . T e GICC_DIR ed deac a e a d a e e ce a ca e , a d a e GICC_DIR e a e ed f deac a SGI :

e SGI N-N a d de

e e a e - e e a ed e e a a de ce.

A a ac e ca deac a e e e f a :

GICV_CTLR.EOImode == 0

T e GIC deac a e a d a e e d ec , a , GICV_EOIR d e f a e a d deac a e a e . T e GICH_LR

G 1 a e a e e ab ed.
G 0 a e a e d ab ed.
G 1 a e a e d ab ed.
T e e a e e d e e L e e .
A ea e EOI e e cc a d L e e e f ec e d e .
T e e a e a d e e , e a d e , e L e e . T a de f c d .
A ea e L e e e a ece ed a EOI e e .

See *M I S R , GICH MISR* a e 5-172 f e f a ab e c a d a e f a e a ce e .

5.2.6 Software-generated interrupts

T e f e f f a e- e e a ed e e :

Hypervisor-generated interrupts

A e ca e e a e a e a d a e a c e d ca e , b c ea a e e L e e e GICH_LR .HW b c ea ed 0. T e e ca c e e a ea a a ac e ead e GICV_IAR GICV_AIAR e e ac ed e e e , b e e e e a :

a SGI, a CPUID a e ded add e e ID
a PPI SPI, e CPUID a e e 0.

T e e ca a e e CPUID a e, b be c e e e f e d ca ed b e GICH_LR .V a ID f e d. W e e EOI f ca e e a CPU e face, e L e e a e affec ed, a d f ca e e D b . See *L R , GICH LR* a e 5-176 f e f a .

Distributor-generated interrupts

Beca e e a d a e e deac a ec a d e SGI , e e a e SGI a f e D b e a e a a e - e e a ed e . T e e ca a e e GICH_LR .CPUID f e d, beca e f e d e ed be e a e a a f e a SGI. See *C* a e 5-161 f e f a ab deac a a ed SGI .

5.2.7 GIC Virtualization Extensions register mapping

I a GIC e e a a c de e V a a E e , e GIC de a a CPU e face, a c e e e f a e face c e e , f eac ce e e . T e GIC a e e e a e face c e e acce b e e f a :

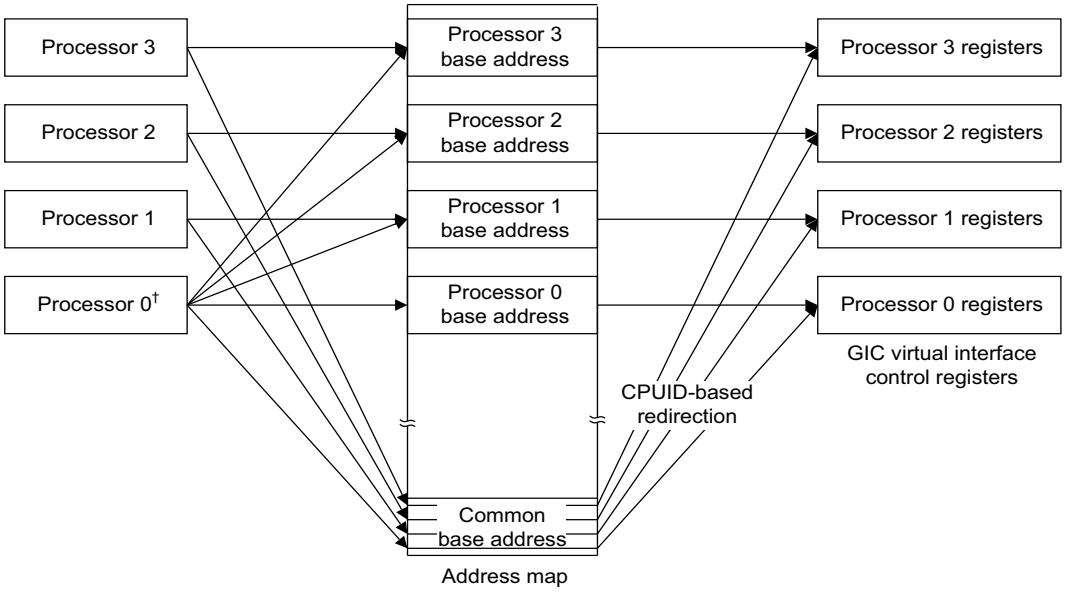
Redirection through a common base address

T e e a c de a f e a e face c e e . Eac ce e e ca acce GIC a e face c e e ba e add e . T e CPUID f e ce e e acce ed ec e acce e GIC a e face c e e f a ce .

Processor-specific base addresses

I add e c ba e add e , e e a c a , f eac ce e e , a ce - ec f c ba e add e f e GIC a e face c e e . A ce ca e e e add e e acce GIC a e face c e e , acce e GIC a e face c e e f a e ce e e .

F e 5-2 a e 5-166 e e a .



† Use of the processor-specific base addresses is shown in full only for accesses by processor 0

Figure 5-2 GIC virtual interface control register mappings

5.3 GIC virtual interface control registers

T e GIC a e face c e e e a e a a e e e e . C f a f a e e ce e e e a e acce b e b a e , a f a e.

Note

A GIC e e a e 32-b de. Re e ed e e add e e a e RAZ/WI.

Tab e 5-1 e e e a f e GIC a e face c e e e .

Table 5-1 GIC virtual interface control register map

Offset	Name	Type	Reset	Description
0x00	GICH_HCR	RW	0x00000000	H e C Re e
0x04	GICH_VTR	RO	IMPLEMENTATION DEFINED	VGIC T e Re e
0x08	GICH_VMCR	RW	IMPLEMENTATION DEFINED	V a Mac e C Re e
0x0C	-	-	-	Re e ed
0x10	GICH_MISR	RO	0x00000000	Ma e a ce I e S a Re e
0x14-0x1C	-	-	-	Re e ed
0x20	GICH_EISR0	RO	0x00000000	E d f I e S a Re e 0 a d 1, ee GICH_EISR
0x24	GICH_EISR1	RO	0x00000000	
0x28-0x2C	-	-	-	Re e ed
0x30	GICH_ELSR0	RO	IMPLEMENTATION DEFINED[a]	E L Re e S a Re e 0 a d 1, ee GICH_ELRSR
0x34	GICH_ELSR1	RO	IMPLEMENTATION DEFINED[a]	
0x38-0xEC	-	-	-	Re e ed
0xF0	GICH_APR	RW	0x00000000	Ac e P e Re e
0xF4-0xFC	-	-	RAZ/WI	Re e ed f GICH_APR1-GICH_APR3
0x100	GICH_LR0	RW	0x00000000	L Re e 0-63, ee GICH_LR
...	-	-	-	
0x1FC	GICH_LR63	RW	0x00000000	

a. Eac b a a ac e d L e e e e 1, ea a e e e a e f e e e de e d e be f L e e e e ed.

5.3.1 Hypervisor Control Register, GICH_HCR

T e GICH_HCR c a ac e c a e:

Purpose T e e c a c b f e a CPU e face.

Usage constraints T e GICH_HCR.E b be e 1 f a a a e a ce e be a e ed.

Configurations T e e a f e GIC V a a E e .

Attributes See e e e a Tab e 5-1 a e 5-167.

F e 5-3 e GICH_HCR b a e .

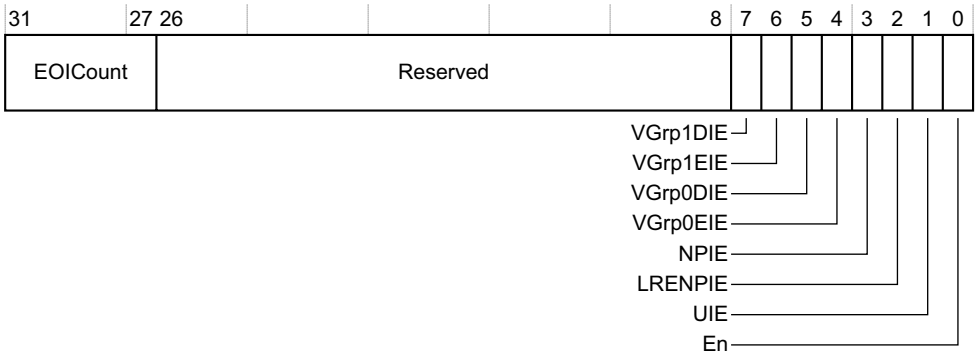


Figure 5-3 GICH_HCR bit assignments

Tab e 5-2 e GICH_HCR b a e .

Table 5-2 GICH_HCR bit assignments

Bit	Name	Description
[31:27]	EOIC	C e be f EOI ece ed a d a e a c e d e e L e e . T e a CPU e face c e e f e d a a ca e a a c EOI ece ed. EOI a d c ea a b e Ac e P e e e, GICH_APR d ca e a c e e . A b e de c ec e a , f a EOI cc e e a e f f e d 31, f e d a 0. T e a e a ce e a e ed e e e f e d - e a d e LRENPIE b e 1.
[26:8]	-	Re e ed.
[7]	VG 1DIE	VM D ab e G 1 I e E ab e. E ab e e a f a e a ce e e a f G 1 e f e a CPU e face e c ec ed a ac e d ab ed: **0** Ma e a ce e d ab ed. **1** Ma e a ce e a ed e GICV_CTLR.E ab e G 1 e 0.
[6]	VG 1EIE	VM E ab e G 1 I e E ab e.

Table 5-2 GICH_HCR bit assignments (continued)

Bit	Name	Description
[5]	VG 0DIE	VM D ab e G 0 I e E ab e. E ab e e a f a a e a ce e e a f G 0 e f e a CPU e face e c ec ed a ac e d ab ed: **0** Ma e a ce e d ab ed. **1** Ma e a ce e a ed e GICV_CTLR.E ab eG 0 e 0.
[4]	VG 0EIE	VM D ab e G 0 I e E ab e. E ab e e a f a a e a ce e e a f G 0 e f e a CPU e face e c ec ed a ac e e ab ed: **0** Ma e a ce e d ab ed. **1** Ma e a ce e a ed e GICV_CTLR.E ab eG 0 e 1.
[3]	NPIE	N Pe d I e E ab e. E ab e e a f a a e a ce e e e d e a e e e e L e e ; **0** Ma e a ce e d ab ed. **1** Ma e a ce e a ed e e L e e c a e e e d a e.
[2]	LRENPIE	L Re e E N P e e I e E ab e. E ab e e a f a a e a ce e e e a CPU e face d e a e a c e d a d L e e e f a EOI e e : **0** Ma e a ce e d ab ed. **1** A a e a ce e a e ed e e EOIC f e d 0.
[1]	UIE	U de f I e E ab e. E ab e e a f a a e a ce e e e L e e a e e , d e a d e : **0** Ma e a ce e d ab ed. **1** A a e a ce e a e ed f e, e, f e L e e e e a ed a a a d e .
[0]	E	E ab e. G ba e ab e b f e a CPU e face: **0** V a CPU e face e a d ab ed. **1** V a CPU e face e a e ab ed. W e f e d e 0: e a CPU e face d e a a a e a ce e e a CPU e face d e a a a e a ead f GICV_IAR GICV_AIAR e a e ID.

T e VG 1DIE, VG 1EIE, VG 0DIE, a d VG 0EIE b e ab e e e ac e a CPU e face a a e e ab ed. T e e ca e e e a a e e a a e a e c ec a d eff c e , a ead e a CPU e face a .

See *M* a e 5-164 a d *M I S R , GICH MISR* a e 5-172 f e f a .

5.3.2 VGIC Type Register, GICH_VTR

T e GICH_VTR c a ac e c a e:

Purpose T a ead- e e a de e f f a ab e e e a f e GIC V a a E e :
be f e e ed
be f ee e e ed
be f e e ed L e e .

Usage constraints T e e a e a e c a .

Configurations T e e a f e GIC V a a E e .

Attributes See e e e a Tab e 5-1 a e 5-167.

F e 5-4 e GICH_VTR b a e .

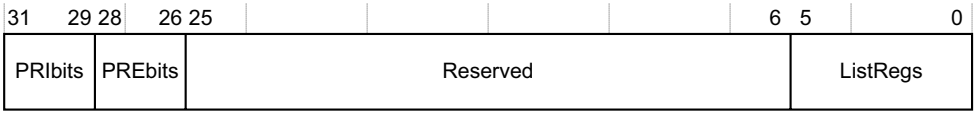


Figure 5-4 GICH_VTR bit assignments

Tab e 5-3 e GICH_VTR b a e .

Table 5-3 GICH_VTR bit assignments

Bit	Name	Description
[31:29]	PRIb	T e be f b e e ed, e. I GIC 2, e a d a e 5 b : **100** 32 e e .
[28:26]	PREb	T e be f ee b e e ed, e. I GIC 2, e a d a e 5 b : **100** 32 ee e e
[25:6]	-	Re e ed, RAZ
[5:0]	L Re	T e be f e e ed L e e , e. F e a e, a a e f `0b111111` d ca e a e a 64 L e e a e e e ed.

5.3.3 Virtual Machine Control Register, GICH_VMCR

T e GICH_VMCR c a ac e c a e:

Purpose E ab e e e a ea d e e a ac e e f e GIC a e.

Usage constraints T e ea e a ec a .

Configurations T e e a f e GIC V a a E e .

Attributes See e e e a Tab e 5-1 a e 5-167.

F e 5-5 e GICH_VMCR b a e .

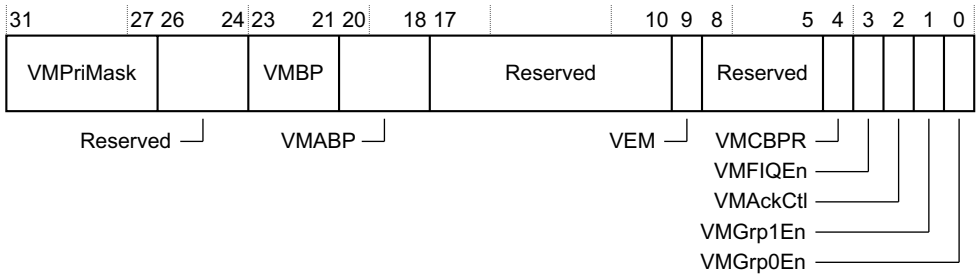


Figure 5-5 GICH_VMCR bit assignments

Tab e 5-2 a e 5-168 e GICH_VMCR b a e .

Table 5-4 GICH_VMCR bit assignments

Bit	Name	Description
[31:27]	VMP Ma	A a f GICV_PMR.P .
[26:24]	-	Re e ed.
[23:21]	VMBP	A a f GICV_BPR.B a . O e e, b e e ed a e f GICV_BPR.
[20:18]	VMABP[a]	A a f GICV_ABPR.B a .
[17:10]	-	Re e ed.
[9]	VEM	A a f GICV_CTLR.EOI de.
[8:5]	-	Re e ed.
[4]	VMCBPR	A a f GICV_CTLR.CBPR.
[3]	VMFIQE	A a f GICV_CTLR.FIQE .
[2]	VMAc C	A a f GICV_CTLR.Ac C .
[1]	VMG 1E	A a f GICV_CTLR.E ab eG 1.
[0]	VMG 0E	A a f GICV_CTLR.E ab eG 0.

a. O e e, e e G 1 b a a e, a , e f VMBP+1, a a ed 7.

T e GICH_VMCR ac e e a c a ead a d e a a e f a c ec e a e e a ac e e , e ab e e a ea d e e a e a e ead e, acce e GIC a CPU e face e e d d a .

5.3.4 Maintenance Interrupt Status Register, GICH_MISR

T e GICH_MISR c a ac e c a e:

Purpose I d ca e c a e a ce e a e a e ed.

Usage constraints A a e a ce e a e ed f a ea e b e e e a d f e ba e ab e b , GICH.HCR.E , e 1.

Configurations T e e a f e GIC V a a E e .

Attributes See e e e a Tab e 5-1 a e 5-167.

F e 5-6 e GICH_MISR b a e .

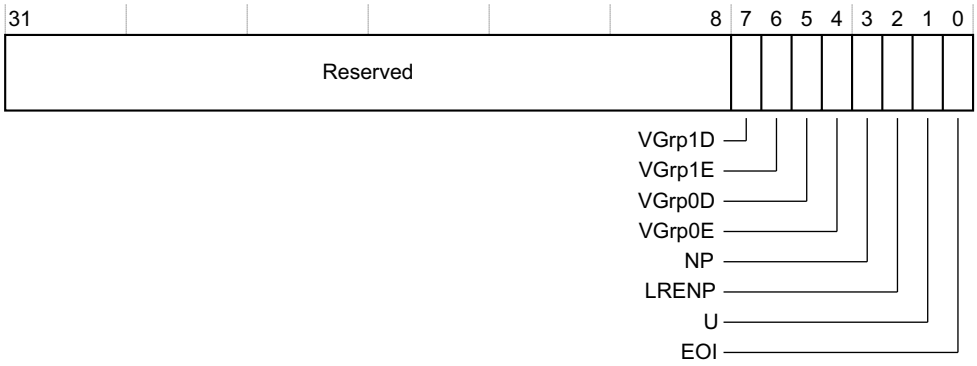


Figure 5-6 GICH_MISR bit assignments

Tab e 5-5 e GICH_MISR b a e .

Table 5-5 GICH_MISR bit assignments

Bit	Name	Description
[31:8]	-	Re e ed.
[7]	VG 1D	D ab ed G 1 a e a ce e . A e ed e e e GICH_HCR.VG 1DIE e a d GICH_VMCR.VMG 1E ==0.
[6]	VG 1E	E ab ed G 1 a e a ce e . A e ed e e e GICH_HCR.VG 1EIE e a d GICH_VMCR.VMG 1E ==1.
[5]	VG 0D	D ab ed G 0 a e a ce e . A e ed e e e GICH_HCR.VG 0DIE e a d GICH_VMCR.VMG 0E ==0.
[4]	VG 0E	E ab ed G 0 a e a ce e . A e ed e e e GICH_HCR.VG 0EIE e a d GICH_VMCR.VMG 0E ==1.
[3]	NP	N Pe d a e a ce e . A e ed e e e GICH_HCR.NPIE==1 a d L e e e d a e.
[2]	LRENP	L Re e E N P e e a e a ce e . A e ed e e e GICH_HCR.LRENPIE==1 a d GICH_HCR.EOIC - e .
[1]	U	U de f a e a ce e . A e ed e e e GICH_HCR.UIE e a d f e, e, f e L e e e e a e a ed a a a d e , a , f e c e d GICH_LR .S a e b d e a 0x0.
[0]	EOI	EOI a e a ce e . A e ed e e e a ea e L e e a e a EOI I e . A ea e b GICH_EISR ==1.

5.3.5 End of Interrupt Status Registers, GICH_EISR0 and GICH_EISR1

T e GICH_EISR c a ac e c a e:

Purpose W e a a e a ce e ece ed, e e e e e de e e c L e e a e a d EOI e a e e e c .

Usage constraints B c e d e e ed L e e a a RAZ.

Configurations T e e e e a e a f e GIC V a a E e . T e be f GICH_EISR de e d e be f L e e e e ed. GICH_EISR0 c e d L e e 0-31 a d GICH_EISR1 c e d L e e 32-63.

Attributes See e e e a Tab e 5-1 a e 5-167.

F e 5-7 e GICH_EISR0 b a e .

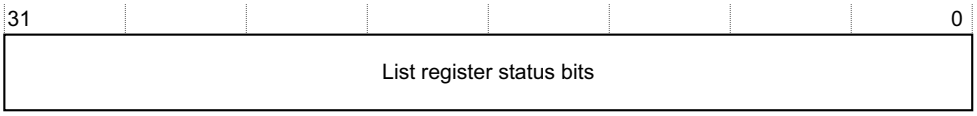


Figure 5-7 GICH_EISR0 bit assignments

Tab e 5-6 e GICH_EISR0 b a e .

Table 5-6 GICH_EISR0 bit assignments

Bits	Name	Function
[31:0]	L e e EOI a b 0-31	F eac b : **0** C e d L e e d e a e a EOI. **1** C e d L e e a a EOI.[a] See *L R , GICH LR* a e 5-176 f e f a .

a. F a GICH_LR , e c e d a b e 1 f (GICH_LR .S a e==00 && GICH_LR .HW ==0 && GICH_LR .EOI==1).

5.3.6 Empty List Register Status Registers, GICH_ELRSR0 and GICH_ELRSR1

T e GICH_ELRSR c a ac e c a e:

Purpose T e e e e ca be ed ca e a ab e L e e e e e de e a e a G e OS.

Usage constraints B c e d e e ed L e e a a RAZ.

Configurations T e e e e a e a f e GIC V a a E e . T e be f GICH_ELRSR de e d e be f L e e e e ed. GICH_ELRSR0 c e d L e e 0-31 a d GICH_ELRSR1 c e d L e e 32-63.

Attributes See e e e a Tab e 5-1 a e 5-167.

F e 5-8 e GICH_ELRSR0 b a e .

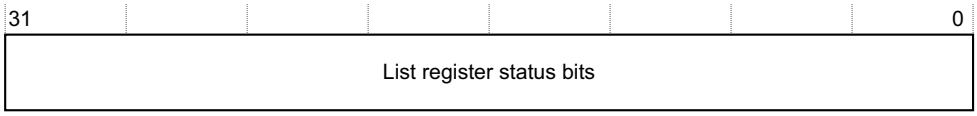


Figure 5-8 GICH_ELRSR0 bit assignments

Tab e 5-7 e GICH_ELRSR0 b a e .

Table 5-7 GICH_ELRSR0 bit assignments

Bits	Name	Function
[31:0]	L e e a b 0-31	F eac b : **0** T ec e d L e e, f e e ed, c a a a d e . U L e e ca e e a a d e . **1** T ec e d L e e d e c a a a d e . T e L e e e a d ca be ed e a a d e a EOI a e a ce e .[a] See *L R , GICH LR* a e 5-176 f e f a .

a. F a GICH_LR , e c e d a b e 1 f (GICH_LR .S a e==00 && (GICH_LR .HW==1 GICH_LR .EOI==0)).

5.3.7 Active Priorities Register, GICH_APR

T e GICH_APR c a ac e c a e:

Purpose T e e ac c ee e e a e ac e e a CPU e face, a d ed de e e e c e ac e . C e d b a e e e e a e ac ed ed, ba ed GICH_LR .P , a d e ea f ca e b c ea ed EOI.

Usage constraints T e b be e de e ed b e f e b f e e .

Configurations T e e a f e GIC V a a E e .

Attributes See e e e a Tab e 5-1 a e 5-167.

F e 5-9 e GICH_APR b a e .

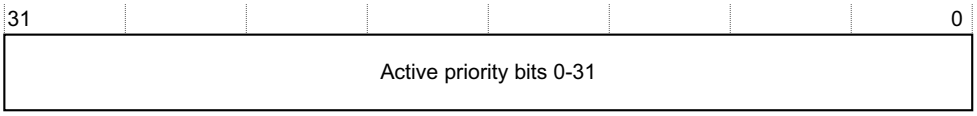


Figure 5-9 GICH_APR bit assignments

Tab e 5-8 e GICH_APR b a e .

Table 5-8 GICH_APR bit assignments

Bit	Name	Description
[31:0]	Ac e b 0-31	De e e e e c e d ee e e ac e: **0** e ee e e ac e **1** e ee e e ac e.

5.3.8 List Registers, GICH_LRn

T e GICH_LR c a ac e c a e:

Purpose P de e c e f a f e a CPU e face.

Usage constraints T e ea e a ec a .

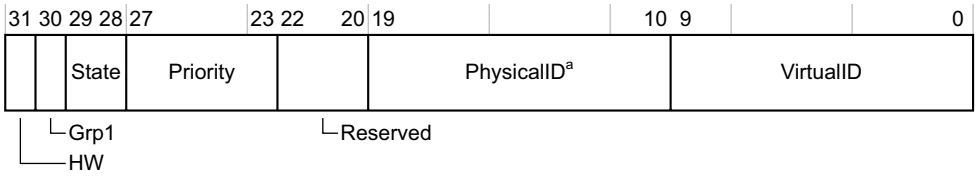
Configurations T e e e a e a f e GIC V a a E e .

A a f 64 L e e ca be ded. T e GICH_VTR.L Re b def e e ac a be e e ed. A e be ed L e e a e RAZ/WI.

A ed b e e a e RAZ/WI.

Attributes See e e e a Tab e 5-1 a e 5-167.

F e 5-10 e GICH_LR b a e .



a These bits have different meaning when GICH_LRn.HW==0.

Figure 5-10 GICH_LR bit assignments

Tab e 5-9 e GICH_LR b a e .

Table 5-9 GICH_LR bit assignments

Bit	Name	Description
[31]	HW	I d ca e e e a e a a d a e e , ea a c e d a ca e . Deac a f e a e a ca e e deac a f e ca e e ID a e P ca ID f e d d ca e . **0** T e e e ed e e f a e. N f ca e e D b e e a e deac a ed. **1** A a d a e e . A deac a e e e e e e D b e e a e deac a ed, b [19:10], e P ca ID, d ca e e ca e ID. If GICV_CTLR.EOI de == 0, e e c e d a e e GICV_EOIR GICV_AEOIR, e e c e d a e e GICV_DIR.
[30]	G 1	I d ca e e e a e a G 1 a e . **0** T a G 0 a e . GICV_CTLR.FIQE de e e e e a ed a a a IRQ a a a FIQ, a d GICV_CTLR.E ab eG 0 e ab e a f e e a ac e. **1** T a G 1 a e , a ed a a a IRQ. GICV_CTLR.E ab eG 1 e ab e e a f e e a ac e. **Note** T e GICV_CTLR.CBPR b c e e GICV_BPR GICV_ABPR ed de e e f a e d G 1 e a ff c e ee c e e ec .

Table 5-9 GICH_LR bit assignments (continued)

Bit	Name	Description
[29:28]	S a e	T e a e f e e . T a e f e f a e : **00** a d **01** e d **10** ac e **11** e d a d ac e. T e GIC da e e e a e b a a e ceed e e fec c e. E e e a d a e a e ed, e ce f e e f e e a a e a ce e . **Note** F a d a e e , e e d a d ac e a e e d e ca D b a e a e a CPU e face. A e e e e d a d ac e a e f f a e a e d e , c a e ca a c a e d a de ce , SGI .
[27:23]	P	T e f e .
[22:20]	-	Re e ed.
[19:10]	P ca ID	T e f c f b de e d e a e f e GICH_LR.HW b , a f . **0** W e GICH_LR.HW e 0, b [19:10] a e e f ea : **[19]** EOI I d ca e e e e a EOI a e a ce e . **0** N a e a ce e a e ed. **1** A a e a ce e a e ed a EOI e e e a e a d, c ca cc e e e deac a ed. **[18:13]** Re e ed, SBZ **[12:10]** CPUID If e e a e V a ID f a SGI, a , 0-15, f e d e e e e CPU ID. T a ea e e e a f e d f e VM I e Ac ed e e e , GICV_IAR GICV_AIAR. O e e, f e d be e 0. **1** W e GICH_LR.HW e 1, f e d d ca e e ca e ID a e e f a d e D b . **Note** W e ed d ca e e ca e ID, f e d e ed e e e b d a a d a e f ec f a ed. A ed e de b a e RAZ/WI. If e a e f P ca ID 0-15, 1020-1023, be a UNPREDICTABLE. If e a e f P ca ID 16-31, f e d a e e PPI a c a e d e a e ca CPUID a e a CPU e face e e e deac a .
[9:0]	V a ID	T ID e ed e G e OS e e e ac ed ed e VM I e Ac ed e e e , GICV_IAR. Eac a d e ed e L e e a e a e V a ID f a a CPU e face. If e a e f V a ID 1020-1023, be a UNPREDICTABLE.

5.4 The virtual CPU interface

A GIC a CPU e face a a e ac ec ed ce , b ec e a GIC a d a d a e . T e GIC a CPU e face e e a e e a e e a f a a e GIC ca CPU e face e e a d e ec ed be a a a a ac e ca d be ee e . T e a e face c e e c a CPU e face e a , a d a c a , e a CPU e face e ec e f e L e e de e e a a e . W e a ce acce e e a CPU e face e L e e a e da ed.

Note

V a e a e a a d ed e a CPU e face .

O e c ec ed ce , f e ce a N - ec e PL1 PL0 de, a e a e a ed e c e a ac e.

I add , a a ac e ca ece e a IRQ a d a FIQ a ed d ec b e e . T e e e ce a e de e c e f ec f ca . A a ac e ca d :

G A a e ce a ed b e GIC f a c e d a e ce a ed d ec b e e .

G A a e ce f e c e d ca e ce .

A a CPU e face d e e e e a a e e , a d e ef e GICV_CTLR d e e e e IRQB D G 1, FIQB D G 1, IRQB D G 0, a d FIQB D G 0 b a a e ed b GICC_CTLR

5.4.1 Enabling and disabling virtual interrupts

T e GICV_CTLR E ab eG 1 a d E ab eG 0 b c e a f G 0 a d G 1 a e e c ec ed a ac e. W e a e a d ab ed, e a CPU e face e a e ID a c e d GICV_IAR GICV_AIAR acce . I IMPLEMENTATION DEFINED e e d ab a e a a e a e effec GICV_HPPIR a d GICV_AHPPIR

W e e ab a d d ab a e e e a , be ece a e e e e e , ee *M* a e 5-164 f e f a ab a c a ed e e .

5.5 GIC virtual CPU interface registers

T e e e e de e a CPU e face acce ed b e a ac e. T ca , a a ac e a a e f a d ffe e ce be ee a e a d ca e . T ea e a e de f a d a e be de ca a f a d ca e . I e e a, e e e e a e e a e f a a e GIC ca CPU e face e e , b e e a e e e e def ed a b e L e e .

T e e e e a e e - a ed, def ed ff e f a IMPLEMENTATION DEFINED GICV_* e e ba e add e .

Note

T e ff e f eac GICV_* e e e a e a e ff e f ec e d e e f e ca CPU e face. F e a e, GICV_PMR a ff e 0x0004 f e GICV_* e e ba e add e , a d GICC_PMR a e a e ff e f e GICC_* e e ba e add e .

T ea a :

- e e ca e e a e 2 add e a a a e a CPU e face acce e e c ec ca add e e .
- f a e, e e acce e e e f a ca CPU e face f a a CPU e face, e e a e e e add e e .

Tab e 5-10 e e e a f e GIC a CPU e face e e .

Table 5-10 GIC virtual CPU interface register map

Offset	Name	Type	Reset	Description
0x0000	GICV_CTLR	RW	0x00000000	V a Mac e C Re e
0x0004	GICV_PMR	RW	0x00000000	VM P Ma Re e
0x0008	GICV_BPR	RW	0x00000002	VM B a P Re e
0x000C	GICV_IAR	RO	0x000003FF	VM I e Ac ed e Re e
0x0010	GICV_EOIR	WO	-	VM E d f I e Re e
0x0014	GICV_RPR	RO	0x000000FF	VM R P Re e
0x0018	GICV_HPPIR	RO	0x000003FF	VM H e P Pe d I e Re e
0x001C	GICV_ABPR	RW	0x00000003	VM A a ed B a P Re e
0x0020	GICV_AIAR	RO	0x000003FF	VM A a ed I e Ac ed e Re e
0x0024	GICV_AEOIR	WO	-	VM A a ed E d f I e Re e
0x0028	GICV_AHPPIR	RO	0x000003FF	VM A a ed H e P Pe d I e Re e
0x002C-0x003C	-	-	-	Re e ed
0x0040-0x00CC	-	-	-	IMPLEMENTATION DEFINED
0x00D0-0x00DC	GICV_APR	RW	IMPLEMENTATION DEFINED	VM Ac e P e Re e
0x00E0-0x00EC	-	-	RAZ/WI	Re e ed f ec d e f Ac e P e Re e , ee e N e e GICV_APR de c .
0x00F0-0x00F8	-	-	-	Re e ed

5.5.1 Virtual Machine Control Register, GICV_CTLR

T e GICV_CTLR c a ac e c a e:

Purpose E ab e a d d ab e G 0 a d G 1 a e .

Note

GICH_LR .G 1 de e e e a a e G 0 G 1.

T e e c e d e GICC_CTLR

Table 5-11 GICV_CTLR bit assignments

Bits	Name	Function
[31:10]	-	Re e ed.
[9]	EOI de	C e be a a c a ed e GICV_EOIR, GICV_AEOIR, a d GICV_DIR e e : **0** GICV_EOIR a d GICV_AEOIR e f d a d deac a e e a a e . GICV_DIR UNPREDICTABLE. W e a c e ed ce e e , e a ac e e GICV_EOIR GICV_AEOIR, deac a e e . T e e: da e e L e e ca e e a CPU e face a e e c e e ca D b . **1** GICV_EOIR a d GICV_AEOIR e f d e a . GICV_DIR e f deac a e e e a . A e d e ce , e a ac e e GICV_EOIR GICV_AEOIR. T e d e f e a e , b da e e L e e . W e a c e ed ce e e , e a ac e e GICV_DIR. T e deac a e e a e a d: da e e L e e ca e e a CPU e face a e e c e e ca D b .
[8:5]	-	Re e ed
[4]	CBPR	C e e e GICV_BPR c b G 0 a d G 1 a e . **0** GICV_BPR c G 0 a e , a d GICV_ABPR c G 1 a e **1** GICV_BPR c G 0 a d G 1 a e . See *T* a e 3-57 f e f a ab GICC_CTLR.CBPR affec acce e GICC_BPR a d GICC_ABPR.
[3]	FIQE	C e e e a ed a G 0 a e e e ed a a FIQ : **0** G 0 e a e e e ed a a IRQ **1** G 0 e a e e e ed a a FIQ .
[2]	Ac C	ARM de eca e e f b . ARM ec e d a f a e e e a e b a a e 0. C e e a ead f e GICV_IAR, e e e e d e a G 1 e , ca e e CPU e face ac ed e e e . **0** If e e e d e a G 1 e , a ead f e GICV_IAR e a I e ID f 1022. T e ead d e ac ed e e e , a d e e d a f e e c a ed. **1** If e e e d e a G 1 e , a ead f e GICV_IAR e e I e ID f e G 1 e . T e ead ac ed e e e , a d e a f e e bec e ac e, ac e a d e d . **Note** O e - e e a ed e ca be ac e a d e d .
[1]	E ab e G 1	E ab e e a f G 1 a e b e a CPU e face e a ac e: **0** S a f G 1 e d ab ed. **1** S a f G 1 e e ab ed.

Table 5-11 GICV_CTLR bit assignments (continued)

Bits	Name	Function
[0]	E ab eG 0	E ab e e a f G 0 a e b e a CPU e face e a ac e: **0** S a f G 0 e d ab ed. **1** S a f G 0 e e ab ed.

5.5.2 VM Priority Mask Register, GICV_PMR

T e GICV_PMR c a ac e c a e:

Purpose P de a a e f e. O a e e a e a e e ca be a ed e ce .

Note

H e c e d a e P f e d a e.

T e P f e d f e e a a ed e VMP Ma f e d GICH_VMCR, e ab e e a e be c ed ea be ee a ac e d c e - c . T e e c e d e GICC_PMR e ca CPU e face.

Usage constraints T e e a e a e c a .

Configurations T e e a f e GIC V a a E e .

Attributes See e e e a Tab e 5-10 a e 5-179.

F e 5-12 e GICV_PMR b a e .

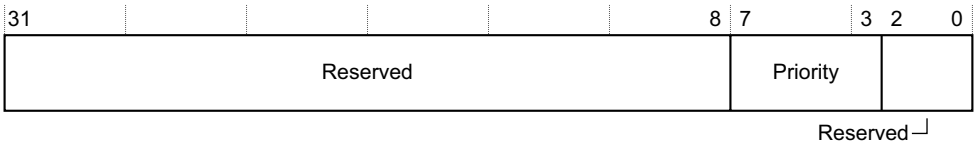


Figure 5-12 GICV_PMR bit assignments

GICV_PMR a GICC_PMR, e c e d e e e GIC ca CPU e face, e ce a b [2:0] a e e e ed. T beca e e a CPU e face fe e a e a e a be f a e a e ca CPU e face ca . See e GICC_PMR de c f e f a ab e b a e .

5.5.3 VM Binary Point Register, GICV_BPR

T e GICV_BPR c a ac e c a e:

Purpose T e e e def e e a c e a e f e d f e a e a , e f e d a d e f e d. T e f e d ed de e e e ee . F e f a ee *P* a e 3-45 a d *P* a e 3-45.

T e e ed de e e e f G 0 e a d, f e GICV_CTLR.CBPR b 1, f G 1 e a . T e e c e d e GICC_BPR e ca CPU e face.

3116.968 611.711T Tf [([(**Usage constraints**)4(4 T e( ).CB 4 abCB 4 a (Tae)B 4de(a) 43)l4 12 CB) 4 fe(e) GICH_VCR)P4Eb -4 GIC de e e 70

5.5.4 VM Interrupt Acknowledge Register, GICV_IAR

Purpose T e a ac e ead e e b a e e ID f e a ed a e . T ead ac a a ac ed e f e e .

T e e c e d e GICC_IAR e ca CPU e face.

Usage constraints T e e a e a e c a .

Configurations T e e a f e GIC V a a E e .

Attributes See e e e a Tab e 5-10 a e 5-179.

F e 5-14 e GICV_IAR b a e .

Figure 5-14 GICV_IAR bit assignments

T e GICV_IAR b a e a e e a ea e b a e f e GICC_IAR, e c e d e e e ca CPU e face. See e GICC_IAR de c f e f a .

W e e ce ead e e , e a CPU e face ac ed e e e e d a e a d e e a e e c e d L e e ac e. T e a a e b e Ac e P e e e , GICH_APR e 1.

If e GICH_LR .HW b e 0, d ca a e e e ed f a e, e b [12:10] f e GICH_LR , a d ca e e CPU ID, a e e ed e GICV_IAR.CPUID f e d. O e e GICV_IAR.CPUID f e d ead a e .

Tab e 5-12 a b e GICV_IAR ead f a a CPU e face.

Table 5-12 Effect of reads of GICV_IAR

Interrupt status	GICV_CTLR.AckCtl	Returned interrupt ID
H e e d e a G 1	1	ID f N - ec e e
	0	I e ID 1022
H e e d e a G 0		ID f Sec e e

Table 5-12 Effect of reads of GICV_IAR (continued)

Interrupt status	GICV_CTLR.AckCtl	Returned interrupt ID
N e d e [a]		I e ID 1023
I e a b a CPU e face d ab ed		I e ID 1023

a. Of ff c e be a ed e ce e a CPU e face e ab ed a d e GICH_HCR.E b e 1

5.5.5 VM End of Interrupt Register, GICV_EOIR

Purpose T e a ac e e e f e a CPU e face a a c e ed e e ce e f e ec f ed a e .
T e e c e d e GICC_EOIR e ca CPU e face.

Usage constraints T e e a e a e c a .

Configurations T e e a f e GIC V a a E e .

Attributes See e e e a Tab e 5-10 a e 5-179.

F e 5-15 e GICV_EOIR b a e .

Figure 5-15 GICV_EOIR bit assignments

T e GICV_EOIR b a e a e a e a e b a e f e GICC_EOIR, e c e d e e e ca CPU e face. See e GICC_EOIR de c f e f a . T ec de c be e e be a f GICV_EOIR d ffe f e be a f GICC_EOIR.

T e be a f GICV_EOIR de e d e e f GICV_CTLR.EOI de:

0 B e d a d e deac a e e effec cc .

1 O e d effec cc .

If e GICH_LR .HW b e a c L e e e 1, d ca a a d a e e , e a deac a e e e e e ca D b , de f e P ca ID f e c e d f e d e L e e . T effec de ca a N - ec e e GICC_DIR f e ce a a ca ID. T ea a f e c e d ca e G 0 e e e ed.

See *B GICC_EOIR, GIC 2* a e 4-140 f e f a .

A cce f EOI e e ea a :

T e e b e GICH_APR c ea ed, ca e d

If GICC_CTLR.EOI de == 0, e e deac a ed e c e d L e e . If e e c e d a a d a e e , e e a deac a ed e D b .

Note

T e e a ca a e e e a e G 1 e a d e ef e G 1 e a e deac a ed e D b .

Tab e 5-13 de a a f GICV_EOIR e a .

Table 5-13 GICV_EOIR operation

Interrupt status	GICV_CTLR.AckCtl	Effect
G 0 e		EOI e a e f ed
G 1 e	0	UNPREDICTABLE
G 1 e	1	EOI e a e f ed

5.5.6 VM Running Priority Register, GICV_RPR

Purpose I d ca e e f e e a e a ac e e a CPU e face.

T e e c e d e GICC_RPR e ca CPU e face.

Usage constraints De e d e e e a , f b a e e e Ac e P e e e , GICH_APR, d ca ac e e e a CPU e face, e ead a `0xFF`, `0xF8` ef ec e be f ed e b , ee *GIC T R , GICH TR* a e 5-170 a d *A P R , GICH APR* a e 5-175.

Configurations T e e a f e GIC V a a E e .

Attributes See e e e a Tab e 5-10 a e 5-179.

F e 5-16 e GICV_RPR b a e .

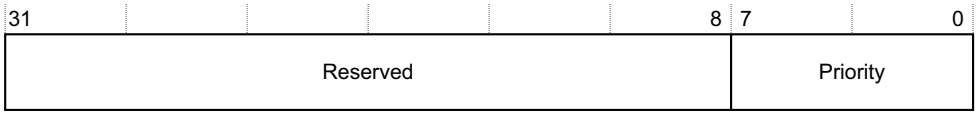


Figure 5-16 GICV_RPR bit assignments

T e GICV_RPR b a e a e e a ea e b a e f e GICC_RPR, e c e d e e e ca CPU e face. See e GICC_RPR de c f e f a .

5.5.7 VM Highest Priority Pending Interrupt Register, GICV_HPPIR

Purpose I d ca e e I e ID f e e d a e e e e a CPU e face. A e e CPU ID f a f a e e , a , f e GICH_LR .HW b ==0.

T e e c e d e GICC_HPPIR e ca CPU e face.

Usage constraints Ne e e e I e ID f a e a ac e a d e d . Re a CPU ID f e e ed f a e.

Configurations T e e a f e GIC V a a E e .

Attributes See e e e a Tab e 5-10 a e 5-179.

F e 5-17 e GICV_HPPIR b a e .

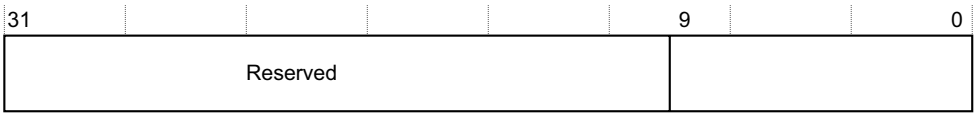


Figure 5-17 GICV_HPPIR bit assignments

T e GICV_HPPIR b a e a e a ea e b a e f e GICC_HPPIR, e c e d e e e ca CPU e face. See e GICC_HPPIR de c f e f a . T ec de c be e be a f GICV_HPPIR d ffe f e be a f GICC_HPPIR.

I ce a a , c a e e e a e e d e a a e ed e L e e , e e e a a a e e a e, 1023. T e ca e a b e beca e:

A e e a e e e

T e e e c ec e e f acc a e e f e e. A e d e e affec e e a e f e e a e ac e.

H e e , a a ee b e , e e a e e a a a a e e e d e e L e e , f e e .

Tab e 5-14 GICV_HPPIR e a .

5.5.8 VM Aliased Binary Point Register, GICV_ABPR

T e GICV_ABPR c a ac e c a e:

Purpose P de a b a e e f e G 1 a e .

Note

GICH_LR .G 1 de e e e a a e G 0 G 1.

T e e c e d e GICC_ABPR e ca CPU e face.

Usage constraints T e a ec a ed e e e ea e a e ac a a ed b a a e, ee *T* a e 3-57 f e f a .

Configurations T e e a f e GIC V a a E e .

Attributes See e e e a Tab e 5-10 a e 5-179.

F e 5-18 e GICV_ABPR b a e .

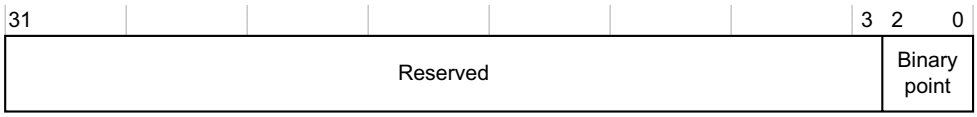


Figure 5-18 GICV_ABPR bit assignments

T e GICV_ABPR b a e a e e a e a e b a e f e GICC_ABPR, e c e d e e e e ca CPU e face. See e GICC_ABPR de c f e f a .

5.5.9 VM Aliased Interrupt Acknowledge Register, GICV_AIAR

T e GICV_AIAR c a ac e c a e:

Purpose A a ac e ead e e b a e e ID f e a ed a e . T ead ac a a ac ed e f a G 1 a e .

O e a a GICV_IAR, e ce a e a ac e e e e ac ed e G 1 e . G 0 e a e ea ed a e .

T e e c e d e GICC_AIAR e ca CPU e face.

Usage constraints T e e a e a e c a .

Configurations T e e a f e GIC V a a E e .

Attributes See e e e a Tab e 5-10 a e 5-179.

F e 5-19 e GICV_AIAR b a e .

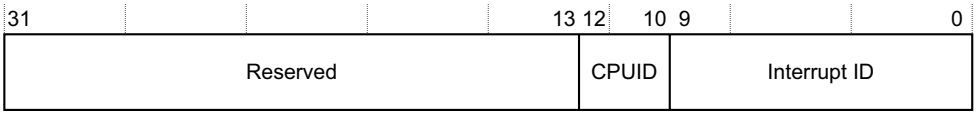


Figure 5-19 GICV_AIAR bit assignments

T e GICV_AIAR b a e a e e a ea e b a e f e GICC_AIAR, e c e d e e e ca CPU e face. See e GICC_AIAR de c f e f a .

T e e a f e e a e e a f GICV_IAR. W e e a ac e ead GICV_AIAR, e c e d e L e e c ec ed de e e e e e e G 0 G 1:

If e GICH_LR .G 1 b 0, e e G 0. T e e ID 1023 e ed a d e e ac ed ed.

If e GICH_LR .G 1 b 1, e e G 1. T e e ID e ed, a d f GICH_LR .HW 0, d ca a e e e e a ed f a e, e CPUID e ed a .

T e L e e e da ed ac e a e, a d e a a e b e GICH_APR, e 1.

If e e e d a e ff c e be a ed e ce , e e e ID 1023 e ed.

Tab e 5-15 GICV_AIAR e a .

Table 5-15 GICV_AIAR operation

Interrupt status	GICV_CTLR.AckCtl	Returned interrupt ID
H e e d e [a] G 0		I e ID 1023
H e e d e G 1		ID f G 1 e
N e d e f ff c e be a ed		I e ID 1023

a. Of ff c e be a ed e ce f e a CPU e face e ab ed a d e GICH_HCR.E b e 1.

5.5.10 VM Aliased End of Interrupt Register, GICV_AEOIR

T e GICV_AEOIR c a ac e c a e:

Purpose A a ac e e e d ca e c e f a G 1 a e . O e a a GICV_EOIR, e ce a e a ac e e e e d ca e c e f G 1 e .

T e e c e d e GICC_AEOIR e ca CPU e face.

Usage constraints T e ea e a e c a .

Configurations T e e a f e GIC V a a E e .

Attributes See e e e a Tab e 5-10 a e 5-179.

F e 5-20 e GICV_AEOIR b a e .

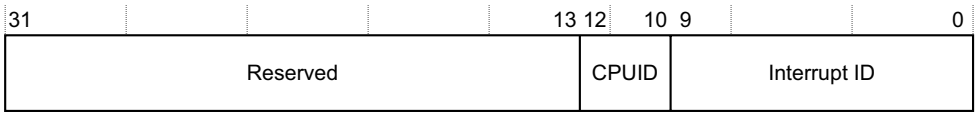


Figure 5-20 GICV_AEOIR bit assignments

A cce f EOI e e ea a :

T e e b e GICH_APR c ea ed, ca e d

If GICC_CTLR.EOI de == 0, e e deac a ed e c e d L e e . If e e c e d a a d a e e , e e a deac a ed e D b .

Note

T e e a ca a e e e a e G 1 e a d e ef e G 1 e a e deac a ed e D b .

Tab e 5-16 GICV_AEOIR e a .

Table 5-16 GICV_AEOIR operation

Interrupt status	GICV_CTLR.AckCtl	Effect
G 0 e		UNPREDICTABLE
G 1 e		EOI e a e f ed

5.5.11 VM Aliased Highest Priority Pending Interrupt Register, GICV_AHPPIR

T e GICV_AHPPIR c a ac e c a e:

Purpose Re e e ID f e e e d G 1 a e e L e e .

T e e c e d e GICC_AHPPIR e ca CPU e face.

Usage constraints Ne e e e I e ID f a e a ac e a d e d .

Configurations T e e a f e GIC V a a E e .

Attributes See e e e a Tab e 5-10 a e 5-179.

F e 5-21 e GICV_AHPPIR b a e .

Figure 5-21 GICV_AHPPIR bit assignments

T e GICV_AHPPIR b a e a e e a ea e b a e f e GICC_AHPPIR, e c e d e e e ca CPU e face. See e *A H P P I R , GICC AHPPIR* a e 4-148 de c f e f a .

Tab e 5-17 GICV_AHPPIR e a .

Table 5-17 GICV_AHPPIR operation

Interrupt status	GICV_CTLR.AckCtl	Returned interrupt ID
H e e d e G 0		I e ID 1023
H e e d e G 1		ID f G 1 e
N e d e f ff c e be a ed		I e ID 1023

5.5.12 VM Active Priorities Registers, GICV_APRn

T e GICV_APR c a ac e c a e:

Purpose F f a ec a b , e e e e a e e e a CPU e face. H e e, a a ac e e ed e e ea d e e a ed e d , a d e ef ed e a e e e e e .

Note

I ead, e e e e GICH_APR e e a e e GIC a e f eac a ac e.

T e e e e c e d e GICC_APR e e e ca CPU e face.

Usage constraints Beca e e e e e a e e ed f e e a d e a e, e c e IMPLEMENTATION DEFINED. Read ec e f e e e e a d e e a e a e d e c a ea a e.

Configurations T e e e e a e a f e GIC V a a E e .

ARM e e e :

GICV_APR0 a a a a f GICH_APR e e a GICV_APR e e a RAZ/WI.

Attributes See e e e a Tab e 5-10 a e 5-179.

T e GICV_APR b a e a e a ea e b a e f e GICC_APR e e , e c e d e e e ca CPU e face. See e GICC_APR de c f e f a .

Note

A a ed ce d e e e a a e Sec e a d N - ec e APR , a d a e e f Ac e P e e e , GICV_APR a e def ed. H e e, e e e a a ca e ace f b e f e e , a e f a ec a b . T e e ace c e d e N - ec e APR RAZ/WI.

5.5.13 VM CPU Interface Identification Register, GICV_IIDR

T e GICV_IIDR c a ac e c a e:

Purpose P de f a ab e e e e a d e f e a CPU e face.
T e e c e d e GICC_IIDR e e e ca CPU e face.

Usage constraints N a e c a .

Configurations T e e a f e GIC V a a E e .

Attributes See e e e a Tab e 5-10 a e 5-179.

F e 5-22 e GICV_IIDR b a e .

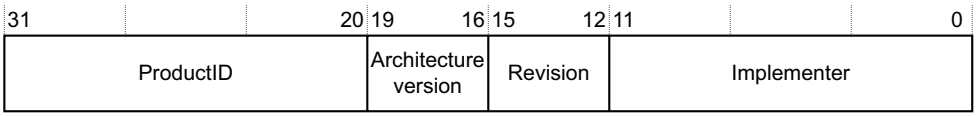


Figure 5-22 GICV_IIDR bit assignments

T e GICV_IIDR b a e a e e a e a e b a e f e GICC_IIDR, e c e d e e e ca CPU e face. See e *CP I I R , GICC IIDR* a e 4-152 de c f e f a .

5.5.14 VM Deactivate Interrupt Register, GICV_DIR

T e GICV_DIR c a ac e c a e:

Purpose T e e deac a e e a e e ec f ed e ID e L e e .

T e e c e d e GICC_DIR e e e ca CPU e face.

Usage constraints W e e e a e a d e GICV_CTLR.EOI de e 1. If GICV_CTLR.EOI de e 0, a e e e UNPREDICTABLE.

Configurations T e e a f e GIC V a a E e .

Attributes See e e e a Tab e 5-10 a e 5-179.

F e 5-23 e GICV_DIR b a e .

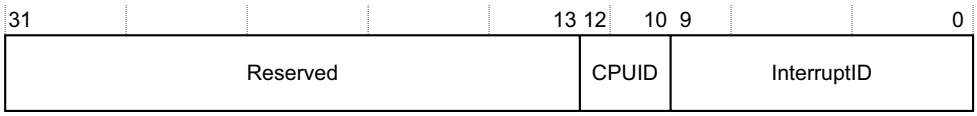


Figure 5-23 GICV_DIR bit assignments

T e GICV_DIR b a e a e e a ea e b a e f e GICC_DIR, e c e d e e e ca CPU e face. See e GICC_DIR de c f e f a . T ec de c be e be a f e GICV_DIR d ffe f e be a f e GICC_DIR.

W e e a ac e e e e, e ec f ed e e L e e c a ed f ac e a d, f ac e a d e d e d . If e ec f ed e e e e L e e e b e ac e e d a d ac e a e, e effec UNPREDICTABLE. If e ec f ed I e d e e e L e e e, e GICH_HCR.EOIc f e d c e e ed, e a e e a a a e a ce e .

Note

If e ec f ed e d e e e L e e e, e a ac e ca ec e e e e ID. T e ef e, e e e e a, e GICV_CTLR.EOI de e 1, e a e ac e e a fe ed f e L e e a f a e . If e a e ac e e a ed e L e e e e, e e a d e acce e GICV_DIR f a e, ca b a e e acce e .

If e GICH_LR .HW b e a c L e e e 1, d ca a a d a e e , e a deac a e e e e e ca D b , de f e P ca ID f e c e d f e d e L e e . T effec de ca a N - ec e e GICC_DIR f e ce a a ca ID. T ea a f e c e d ca e a ed a G 0, e e e ed.

Note

I e deac a GICV_DIR ba ed e ded e ID, e e e deac a e e a a c a de . A e e e e ef e ed deac a e G 0 a d G 1 e .

Pseudocode Index

T a e d e a de f e e d c de f c def ed ec f ca . I c a e f ec :

I a e A-198.

A.1 Index of pseudocode functions

Tab e A-1 a de f e e d c de f c def ed ec f ca . W e e d ffe e f f e f c a e ed e a c ec e a d e Sec E e , e de efe b f .

Note

T e e d c de d c e f e ARM a c ec e e d c de c e . F e f a , ee *ARM A R M ARM 7-A ARM 7-R* .

Table A-1 Pseudocode functions and procedures

Function	Meaning	See
`AcknowledgeInterrupt()`	Se e ac e a e a d a e c ea e e d a e f e e a c a ed a e I e ID.	*G* a e 3-61
`AnyActiveInterrupts()`	Re TRUE f a e e ac e a e.	*G* a e 3-61
`BinaryPointRegWrite()`	W e be a f acce e GICC_BPR e e Sec E e a e e e ed.	*B P R , GICC BPR* a e 4-133
`GIC_GenerateExceptions()`	E ce e e a b e CPU e face e GIC a c e e.	*E* a e 3-64 *E* , a e 3-64
`GIC_PriorityMask()`	Re e a be ed f a a f e a	*G* a e 3-61
`HighestPriorityPendingInterrupt()`	Re e ID f e e a e d . If e a e e d , e a e ID.	*G* a e 3-61
`IgnoreWriteRequest()`	N e a . I d ca e ca e e e e GIC e a e a e e.	*G* a e 3-61
`IsEnabled()`	Re TRUE f e e e ab ed.	*G* a e 3-61
`IsGrp0Int()`	Re TRUE f e e de f ed b e f c a e c f ed a a G 0 e .	*G* a e 3-61
`IsPending()`	Re TRUE f e e de f ed b e f c a e e d .	*G* a e 3-61
`MaskRegRead()`	Read be a f acce e GICC_PMR e e Sec E e a e e e ed.	*I P M R , GICC PMR* a e 4-131
`MaskRegWrite()`	W e be a f acce e GICC_PMR e e Sec E e a e e e ed.	*I P M R , GICC PMR* a e 4-131

Table A-1 Pseudocode functions and procedures (continued)

Function	Meaning	See
`PriorityIsHigher()`	Re TRUE f e f a e f e f c a a e a e ec d a e .	*G* a e 3-61
`PriorityRegRead()`	Read be a f acce e e GICD_IPRIORITYR , GICC_PMR a d GICC_RPR e e Sec E e a e e e ed.	*T* *GIC S E* a e 3-66
`PriorityRegWrite()`	W e be a f acce e e GICD_IPRIORITYR a d GICC_PMR e e Sec E e a e e e ed.	*T* *GIC S E* a e 3-66
`ReadGICC_HPPIR()`	Re e a e f GICC_HPPIR ead b a CPU acce .	*H P P I R* , *GICC HPPIR* a e 4-143
`ReadGICC_IAR()`	Re e a e f GICC_IAR ead b a CPU acce .	*I A R* , *GICC IAR* a e 4-135
`ReadGICC_RPR()`	Re e a e f GICC_RPR ead b a CPU acce .	*R P R* , *GICC RPR* a e 4-142
`ReadGICD_IPRIORITYR()`	Re e a e f e e de f ed b e f c a e , b ead e a a e GICD_IPRIORITYR .	*G* a e 3-61
`ReadGICD_ITARGETSR()`	Re a 8-b f e d ec f c ce a e ece e e e ec f ed b a e I e ID.	*G* a e 3-61
`SGI_CpuID()`	Re e ID f e e ce f e f a e e e a ed e ec f ed b I e ID.	*G* a e 3-61
`SignalFIQ()`	If e a a e e TRUE, a e a e ce e e a FIQ e ce .	*G* a e 3-61
`SignalIRQ()`	If e a a e e TRUE, a e a e ce e e a IRQ e ce .	*G* a e 3-61
`UpdateExceptionState()`	GIC e ce a c e e ed b e CPU e face	*E* a e 3-64
`WriteGICD_IPRIORITYR()`	Se e a e f e e de f ed b e f c a e , b e a a e GICD_IPRIORITYR .	*G* a e 3-61

Register Names

T a e d de c be e e a be ee e a c ec a a e f e e e de c bed
ec f ca , a d e e ac a a e . I a de a de f e a c ec a a e . I c a e f
ec :

A a e B-202

R a e B-203

I a e B-204.

B.1 Alternative register names

GIC 2 e e ace e e e a e f GIC 1 e e . Tab e B-1 e GIC 1 a e a d e GIC 2 e ed e ace e a e f e e e e D b .

Table B-1 Replacement names for the registers in the Distributor

Register	GICv2 name	GICv1 name
D b C	GICD_CTLR	ICDDCR
I e C e T e	GICD_TYPER	ICDICTR
D b I e e e Ide f ca	GICD_IIDR	ICDIIDR
I e G	GICD_IGROUPR	ICDISR
I e Se -Ac e	GICD_ISACTIVER	ICDABR
I e Se -E ab e	GICD_ISENABLER	ICDISER
I e C ea -E ab e	GICD_ICENABLER	ICDICER
I e Se -Pe d	GICD_ISPENDR	ICDISPR
I e C ea -Pe d	GICD_ICPENDR	ICDICPR
I e P	GICD_IPRIORITYR	ICDIPR
I e P ce Ta e	GICD_ITARGETSR	ICDIPTR
I e C f a	GICD_ICFGR	ICDICR
S f a e Ge e a ed I e	GICD_SGIR	ICDSGIR
Ide f ca	-	-

Tab e B-2 e GIC 1 a e a d e GIC 2 e ed e ace e a e f e e e e CPU e face.

Table B-2 Replacement names for the registers in the CPU interface

Register	GICv2 name	GICv1 name
CPU I e face C	GICC_CTLR	ICCICR
P Ma	GICC_PMR	ICCPMR
B a P Re e	GICC_BPR	ICCBPR
I e Ac ed e	GICC_IAR	ICCIAR
E d f I e	GICC_EOIR	ICCEOIR
R P	GICC_RPR	ICCRPR
A a ed B a P	GICC_ABPR	ICCABPR
H e P Pe d I e	GICC_HPPIR	ICCHPIR
CPU I e e e Ide f ca	GICC_IIDR	ICCIIDR

B.2 Register name aliases

S e e e a f GIC a c ec e, f ca ea , d e ea c ec a a e f e e e de c bed ec f ca . De e e e d c be a a ed f e e f e ARM GIC a c ec e. Tab e B-3 e a a a e a a e e e ed f e e e e D b .

Table B-3 Alias names for the registers in the Distributor

Register	Name	Alias
D b C	GICD_CTLR	e ab e_ , e ab e_
I e C e T e	GICD_TYPER	c_ e_ e
D b I e e e Ide f ca	GICD_IIDR	d _ de _ e
I e G	GICD_IGROUPR	_ ec
I e Se -E ab e	GICD_ISENABLER	e ab e_ e
I e C ea -E ab e	GICD_ICENABLER	e ab e_c
I e Se -Pe d	GICD_ISPENDR	e d _ e
I e C ea -Pe d	GICD_ICPENDR	e d _c
I e P	GICD_IPRIORITYR	_ e e
I e P ce Ta e	GICD_ITARGETSR	a e
I e C f a	GICD_ICFGR	_c f
S f a e Ge e a ed I e	GICD_SGIR	_c
Ide f ca	-	-

Tab e B-4 e a a a e a a e e e ed f e e e e CPU e face.

Table B-4 Alias names for the registers in the CPU interface

Register	Name	Alias
CPU I e face C	GICC_CTLR	c _ , c _
P Ma	GICC_PMR	_ a
B a P Re e	GICC_BPR	b _ _ , b _ _
I e Ac ed e	GICC_IAR	_ac
E d f I e	GICC_EOIR	EOI
R P	GICC_RPR	_
A a ed B a P	GICC_ABPR	a a _b _ _
H e P Pe d I e	GICC_HPPIR	_ e d
CPU I e e e Ide f ca	GICC_IIDR	c _ de

B.3 Index of architectural names

Tab e B-5 a a abe c de f e GIC e e a e , de e de c f eac e e . A a e e d f a e e a e, a GICC_APR , a e e a e e a ce f e e e .

Table B-5 Index of GIC register names

Register name	Description
C e ID	*I* a e 4-119
GICC_ABPR	*A B P R , GICC ABPR* a e 4-145
GICC_APR	*A P R , GICC APR* a e 4-149
GICC_AEOIR	*A E I R , GICC AEOIR* a e 4-147
GICC_AIAR	*A I A R , GICC AIAR* a e 4-146
GICC_AHPPIR	*A H P P I R , GICC AHPPIR* a e 4-148
GICC_BPR	*B P R , GICC BPR* a e 4-133
GICC_CTLR	*CP I C R , GICC CTLR* a e 4-125
GICC_DIR	*D I R , GICC DIR* a e 4-153
GICC_EOIR	*E I R , GICC EOIR* a e 4-138
GICC_HPPIR	*H P P I R , GICC HPPIR* a e 4-143
GICC_IAR	*I A R , GICC IAR* a e 4-135
GICC_IIDR	*CP I I R , GICC IIDR* a e 4-152
GICC_NSAPR	*N - A P R , GICC NSAPR* a e 4-151
GICC_PMR	*I P M R , GICC PMR* a e 4-131
GICC_RPR	*R P R , GICC RPR* a e 4-142
GICD_CPENDSGIR	*SGI C -P R , GICD CPENDSGIR* a e 4-115
GICD_CTLR	*D C R , GICD CTLR* a e 4-85
GICD_ICACTIVER	*I C -A R , GICD ICACTI ER* a e 4-103
GICD_ICENABLER	*I C -E R , GICD ICENABLER* a e 4-95
GICD_ICFGR	*I C R , GICD ICFGR* a e 4-109
GICD_ICPENDR	*I C -P R , GICD ICPENDR* a e 4-99
GICD_IGROUPR	*I G R , GICD IGRO PR* a e 4-91
GICD_IIDR	*D I I R , GICD IIDR* a e 4-90
GICD_IPRIORITYR	*I P R , GICD IPRIORIT R* a e 4-104
GICD_ISACTIVER	*I S -A R , GICD ISACTI ER* a e 4-102
GICD_ISENABLER	*I S -E R , GICD ISENABLER* a e 4-93
GICD_ISPENDR	*I S -P R , GICD ISPENDR* a e 4-97
GICD_ITARGETSR	*I P T R , GICD ITARGETSR* a e 4-106

Table B-5 Index of GIC register names (continued)

Register name	Description
GICD_SGIR	*S G I R , GICD SGIR* a e 4-113
GICD_NSACR	*N - A C R , GICD NSACR* a e 4-111
GICD_SPENDSGIR	*SGI S -P R , GICD SPENDSGIR* a e 4-117
GICD_TYPER	*I C T R , GICD T PER* a e 4-88
Pe e a ID	*I* a e 4-119
GICH_APR	*A P R , GICH APR* a e 5-175
GICH_EISR	*E I S R , GICH EISR0 GICH EISR1* a e 5-173
GICH_ELRSR	*E L R S R , GICH ELRSR0 GICH ELRSR1* a e 5-174
GICH_HCR	*H C R , GICH HCR* a e 5-168
GICH_LR	*L R , GICH LR* a e 5-176
GICH_MISR	*M I S R , GICH MISR* a e 5-172
GICH_VMCR	*M C R , GIC CTLR* a e 5-180
GICH_VTR	*GIC T R , GICH TR* a e 5-170
GICV_ABPR	*M A B P R , GIC ABPR* a e 5-190
GICV_AEOIR	*M A E I R , GIC AEOIR* a e 5-192
GICV_AHPPIR	*M A H P P I R , GIC AHPPIR* a e 5-193
GICV_AIAR	*M A I A R , GIC AIAR* a e 5-191
GICV_APR	*M A P R , GIC APR* a e 5-194
GICV_BPR	*M B P R , GIC BPR* a e 5-184
GICV_CTLR	*M C R , GIC CTLR* a e 5-180
GICV_EOIR	*M E I R , GIC EOIR* a e 5-187
GICV_HPPIR	*M H P P I R , GIC HPPIR* a e 5-189
GICV_IAR	*M I A R , GIC IAR* a e 5-185
GICV_PMR	*M P M R , GIC PMR* a e 5-183
GICV_RPR	*M R P R , GIC RPR* a e 5-188
GICV_IIDR	*M CP I I R , GIC IIDR* a e 5-195
GICV_DIR	*M D I R , GIC DIR* a e 5-196

Revisions

T a e d de c be e a ec ca c a e be ee e ea ed e f b . T e e a e ec ca c a e be ee e B a d e B.b, ee e S a a e e e Re ea e I f a a e a f e b .

Table C-1 Differences between issue A and issue B

Change	Location
Sec da ed de c be e f c a	*A G I C* a e 1-14
Sec da ed	*C 2.0 S* a e 1-15
Sec da ed	*S E* a e 1-16
Sec added	a e 1-17
Sec da ed c a f SGI de c a d c de a e	*I* a e 1-18
Sec da ed de c be a CPU e face	*A GIC* a e 2-22
N e added c a f GIC 1 f c a	*T D* a e 2-24
Sec da ed c a f GIC 2 CPU e face be a	*CP* a e 2-26
Sec added	*I , GIC 2* a e 2-27
Sec added	*P , GIC 2* a e 2-31
GICC_CTLR.SBPR b e a ed GICC_CTLR.CBPR c a f e .	C a e 3 *I H P* C a e 4 *P M*

Table C-1 Differences between issue A and issue B (continued)

Change	Location
Sec da ed c de e f c a	*A* a e 3-34 *I* a e 3-35
Sec da ed c a f EOI be a a d c e.	*G* a e 3-37
Sec added	*P* a e 3-38
Sec da ed c a f EOI be a a d f Sec e e GICD_SGIR	*I* a e 3-41
Sec da ed c a f c e f e f c a	*I* a e 3-44
Sec da ed c a f ee be a a d c de f a ab d f c a	*P* a e 3-45
Sec da ed c a f f c a	*P* a e 3-45
Added ab e c a f be a	Tab e 3-3 a e 3-46
Sec e a ed a d da ed c a f c e, a d de c be e f c a	*T* a e 3-48
Sec da ed c a f e a d	*T* a e 3-50
Sec added	*GIC* a e 3-51
Sec e a ed a d da ed c a f c e, a d de c be e f c a	*I* a e 3-53
Sec da ed c a f f c a	*T* a e 3-57
Sec added	*A* *GIC S* *E* a e 3-59
Sec added	*A* *ARM S* *E* a e 3-59
P e d c de da ed	*P* a e 3-61
Sec added	*T* *E* a e 3-67
Sec added	*E* *GIC* a e 3-68
D b a d CPU e face e e a ab e da ed	Tab e 4-1 a e 4-75 Tab e 4-2 a e 4-76
N e added c a f e d a e	*GIC* a e 4-77
Sec da ed c a f e e ba ce e	*R* a e 4-77
Sec added	*E* *D* *CP* a e 4-77
Sec da ed c de e f c a	*E* *GIC S* *E* a e 4-80

Table C-1 Differences between issue A and issue B (continued)

Change	Location
Sec da ed de c be GIC 1 a d GIC 2 d ffe e ce a d e effec f e Sec E e	*D C R , GICD CTLR* a e 4-85
Re e e a ed f I e Sec Re e , a d ec da ed c a f c e	*I G R , GICD IGRO PR* a e 4-91
Sec da ed c a f e e de c	*I S -E R , GICD ISENABLER* a e 4-93 *I C -E R , GICD ICENABLER* a e 4-95
Sec da ed c a f e e - e e e f a	*I C -P R , GICD ICPENDR* a e 4-99
D b e e de c added	*I S -A R , GICD ISACTI ER* a e 4-102 *I C -A R , GICD ICACTI ER* a e 4-103 *N - A C R , GICD NSACR* a e 4-111 *SGI C -P R , GICD CPENDSGIR* a e 4-115 *SGI S -P R , GICD SPENDSGIR* a e 4-117
P e d c de added e effec f e GIC Sec E e acce e e e e e	*I P R , GICD IPRIORIT R* a e 4-104
Sec da ed c a f a e c a a d c a e b a e.	*S G I R , GICD SGIR* a e 4-113
Sec da ed de c be effec f GIC 2	*I* a e 4-119
Sec da ed de c be GIC 1 a d GIC 2 e e a d e effec f e Sec E e	*CP I C R , GICC CTLR* a e 4-125
P e d c de added e effec f e GIC Sec E e acce e e e	*I P M R , GICC PMR* a e 4-131
Sec da ed c de e f c a	*B P R , GICC BPR* a e 4-133
Sec da ed c a f e ac ed e e be a	*I A R , GICC IAR* a e 4-135
Sec da ed c a f EOI be a	*E I R , GICC EOIR* a e 4-138
Sec da ed de c be effec f Sec E e	*B GICC EOIR, GIC 1 S E* a e 4-139
Sec added	*B GICC EOIR, GIC 2* a e 4-140
Sec da ed c a f effec f e f c a	*H P P I R , GICC HPPIR* a e 4-143
P e d c de added e effec f e GIC Sec E e acce e e e	
Sec da ed c a f be a	*A B P R , GICC ABPR* a e 4-145

Table C-1 Differences between issue A and issue B (continued)

Change	Location
CPU e face e e de c added	*A I A R , GICC AIAR* a e 4-146 *A E I R , GICC AEOIR* a e 4-147 *A H P P I R , GICC AHPPIR* a e 4-148 *A P R , GICC APR* a e 4-149 *N - A P R , GICC NSAPR* a e 4-151 *D I R , GICC DIR* a e 4-153
Sec added	*P GIC* a e 4-155
C a e added	C a e 5 *GIC S*
A e d e ed[a]	A e d B *S E GIC*
Sec added	*A* a e B-202
Sec da ed c de GIC 2 e e	*I* a e B-204

a. T c e de e c e f e A c ec e S ec f ca .

Glossary

Activate A e ac a ed e a e c a e e e :
f e d ac e
f e d ac e a d e d .

F e f a ee *I* a e 3-41.

Banked interrupt
I a ce e e a , a ba ed e e f e PPI SGI a a e e a e e ID, b a e d ffe e c ec ed ce a d a e de e de a e c e d eac c ec ed ce .

Banked register A e e a a e a ce .A e f e a e f e de ce de e e c a ce e. F e f a ab e e ba e GIC ee *R* a e 4-77.

Deactivate A e deac a ed e a e c a e e e :
f ac e ac e
f ac e a d e d e d .

F e f a ee *I* a e 3-41.

Idle priority T e e b e a ca be a ed a e .I a e e a a e -b f e d , e a e f e d e `0xFF`. O e e, e e e a e a e c a RW GICD_IPRIORITYR .P f e d ca be a ed, `0xFF`.

IMP I a abb e a ed d a a d ca e a e b b c ce ed a e IMPLEMENTATION DEFINED be a .

IMPLEMENTATION DEFINED
Mea a e be a a c ec a def ed, b d be def ed a d d c e ed b d d a e e a .

IMPLEMENTATION SPECIFIC
Mea a e be a a c ec a def ed, a d d e a e be d c e ed b d d a

e e a . U ed e e e a e a be f e e a a a ab e a d e c e d e affec f a e c a b .

Interrupt grouping

T e c f a f e a e e G 0 G 1. O e e f e a a e Sec e a d N - ec e e , G 0 f Sec e e a d G 1 f N - ec e e .

Local access A ca acce a a c a GIC a acce f a ce a CPU e face a GIC. Re e a d ca acce e ed SPI , b SGI ca acce . See a Re e acce .

Observer

A ce ec a e e , c a e e a de ce, a ca ab e f e e a ead f e e .

Peripheral interrupt

A e e e a ed b e a e f a e e e a e GIC. T e GIC a c ec e def e e f e f e e a e :

Private Peripheral Interrupt (PPI)

A e e a e a ec f c a e ce .

Shared Peripheral Interrupt (SPI)

A e e a e a e D b ca e ac b a f ce , a ec f ed b e c e d GICD_ITARGETSR e e .

PPI *S* Pe e a I e

Preemption level

A ee e e a ed . F e f a , ee *P* a e 3-45.

Priority drop P d e e f a CPU e face e e f e ece ac ed ed ac e e , a a bee b ec a EOI e e , b e e e a ac e.

S R .

RAO *S* Read-A -O e.

RAO/WI Read-A -O e, W e I ed.I a e e a , e b ead a 1, a 1 f ab f e d, a d e e f e d be ed.

S f a e ca e e b ead a 1, a 1 f ab f e d, a d e be ed.

RAZ *S* Read-A -Ze .

RAZ/WI Read-A -Ze , W e I ed.I a e e a , e b ead a 0, a 0 f ab f e d, a d e e f e d be ed.

S f a e ca e e b ead a 0, a 0 f ab f e d, a d e be ed.

Read-As-One (RAO)

I a e e a , e b ead a 1, a 1 f ab f e d.

Read-As-Zero (RAZ)

I a e e a , e b ead a 0, a 0 f ab f e d.

Remote access A e e acce a a c a GIC a acce f a ce a CPU e face a GIC. Re e a d ca acce e ed SPI , b SGI ca acce . See a L ca acce .

Reserved Re e a a e e e ed a e RAZ/WI e e e a ed. B de c be d a Re e ed a e UNK/SBZP.

Running priority

T e f a CPU e face e e :

e f e e ac e e , a e face, f c e e a bee a a d e a e d f e e e

f e e ac e e e face f c e e a bee a a d e a e d f e e e, e e d e .

S Id e , *P* a e 3-38.

SBZ *S* S d-Be-Ze .

SBZP *S* S d-Be-Ze - -P e e ed.

Security hole I a ec a a b a e e ec .

SGI *S* S f a e- e e a e d e .

Should-Be-Zero (SBZ)
S d be e a 0 (a 0 f a b f e d) b f a e. Va e e a 0 d ce UNPREDICTABLE e .

Should-Be-Zero-or-Preserved (SBZP)
M be e a 0, a 0 f a b f e d, b f a e f e a e be e a bee e ead, f e e e a bee a e d. W e e e e e a e ead e a e ce , ce e ce a a e e, e a e e f e d d be e e e d b e a e a a e ead.

Ha d a e e e e e f e d .

If a a e e e f e d a e e 0 (a 0 f a b f e d), a a e e ead f e a e f e d e a e ce , e e UNPREDICTABLE.

Software-generated interrupt (SGI)
A e e e a e d b e GIC e e f a e a GIC e e . I a ce e e a , a SGI de f e d b e c b a f e ID a d e CPU ID f e ce a e e GIC e e a e e e .

SPI *S* Pe e a I e

Spurious interrupt
A e a d e e e e c . U a , e f e a e ID e e d b a GIC a e e f a c e c e d ce . Re a e ID d c a e a e e e d e e CPU e face a e e e ce ca e ce. F e a e, f a e e - e e e e e a e GIC ca e a CPU e face a a e e e a ce , b b e e e ce ead e GICC_IAR ac ed e e e e e e a a bee dea e ed, e GIC e a e ID f 1023, d c a e a e e e e e e ce.

Sufficient priority
T de e e e e a a e c e c ed ce , a GIC CPU e face de e e e e e e a be a e d e c e c ed ce . I d e b c a e e a f:

- e P Ma Re e , GICC_PMR
- e ee e f e e face, a b GICC_BPR GICC_ABPR
- e c e f e CPU e face.

If e e a ff c e e a e e e a e d e c e c ed ce .

S R .

UNK S f a e ea a f e d a c a a UNKNOWN a e. I a e e a , e b ead a 0, a 0 f a b f e d. S f a e e e f e d ead a e .

UNKNOWN A UNKNOWN a e d e c a a d d a a, a d c a a f e e , c c , a d e e a e e a . A UNKNOWN a e be a ec e. UNKNOWN a e be d c e e d ed a a a def e d a e effec .

UNK/SBZP UNKNOWN ead , S d-Be-Ze - -P e e ed e .

I a e e a , e b ead a 0, a 0 f a b f e d, a d e e f e d be e d.

S f a e e e f e d ead a 0, a 0 f a b f e d, a d e a SBZP c e e e f e d.

UNPREDICTABLE

T e be a ca be e ed . UNPREDICTABLE be a e e e ec e . UNPREDICTABLE be a a a e ce , a a f e e . UNPREDICTABLE be a be d c e ed ed a a a def ed effec .

Valid interrupt ID

A e ID, a e ed b a ead f GICC_IAR GICC_AIAR, a a e ID. T ea a e ID a a e f 1019 e . If e e a SGI, e e ec e d ca e e e, e a d e ID c de e a c a ed CPUID.