

- 10.

10.

- ARM® Cortex™-A Series Programmer’s Guide version4.0
- ARM® Architecture Reference Manual ARMv7-A and ARMv7-R edition
- Cortex™-A7 MPCore™ Technical Reference Manual Revision: r0p5
- ARM® Generic Interrupt Controller Architecture Specification Architecture version 2.0
- Chapter 3 & Chapter28 : Ultra Secured Digital Host Controller (uSDHC)

ARM® Cortex™-A Series Programmer’s Guide version4.0

10.1 ARM



ARM

7

TrustZone

monitor mode



TrustZone

ARMv7-A

hypervisor mode Hyp

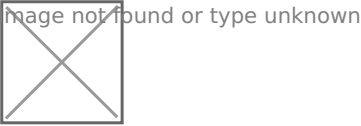
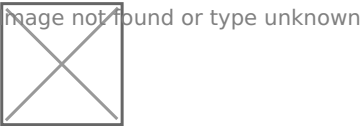
ARM



PL0 PL1 PL2

PL0

- PL0 PL0
- PL1 PL1 Hyp PL1 PL0
- PL2 hypervisor PL2 Hyp mode PL1 guest OS hypervisor PL2 Hyp

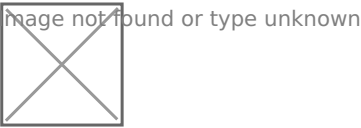


TrustZone

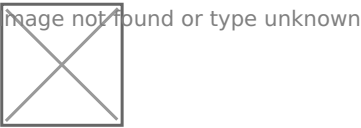
Linux

CPSR

10.1.1



ARM 32 R0-R15 0 15 R0-R14 R15 R15



“ ” R15 3-5 “ ” FIQ R8-R12 FIQ

R13 R13 R13_usr SVC R13_svc

R13

R14 BL R14_svc R14_irq R14_fiq R14_abt R14_und

R15 ARM Thumb ARM1

| | | | | | | |
|------------|-----|------------|-------|-----|--------|-------|
| R0-R14 | SP | ARM | AAPCS | ARM | ABI | AEABI |
| Hypervisor | | | | | | |
| | Hyp | Hypervisor | PL2 | R13 | SP_hyp | SPSR |

10.1.2

CPSR current programmer status register



bits[31:28]

N bit[31]

Z bit[30]

C bit[29]

V bit[28]

GE[3:0],bit[19:16] SIMD

IT[7:2] bit[15:10]Thumb2 If-then

J, bit[24] Jazelle T, bit[5]

E, bit[9] 0 1

Mask bits, bits[8:6]

A, bit[8] Asynchronous abort mask bit.

I, bit[7] IRQ

F, bit[6] FIQ

Q bit[27] 1 DSP

T, bit[5] Thumb J bit[24] ARM Thumb Jazelle ThumbEE

M[4:0], bits[4:0]

CPSR

PSR [4 0] A I F

10.1.3 CP15

CP15 16 32 CP15 CP15

Rt CP15 CRn Op1 Op2 CRm

CP15 ARM

MRC p15, Op1, Rt, CRn, CRm, Op2 ; read a CP15 register into an ARM register

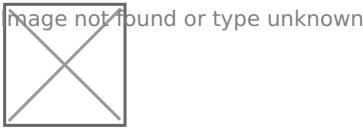
ARM CP15

MCR p15, Op1, Rt, CRn, CRm, Op2 ; write a CP15 register from an ARM register

10.1.3.1 System control register (SCTLR)

SCTLR CP15

PL1



- TE -Thumb ARM Thumb
- NMFI - FIQ NMFI
- EE = CPSR.E
- U -
- FI - FIQ
- V -
- I -
- Z -

- C –
- A –
- M – MMU

CP15 SCTLR Z

MRC p15, 0, r0, c1, c0, 0 ; Read System Control Register configuration data

ORR r0, r0, #(1 << 2) ; Set C bit

ORR r0, r0, #(1 << 12) ; Set I bit

ORR r0, r0, #(1 << 11) ; Set Z bit

MCR p15, 0, r0, c1, c0, 0 ; Write System Control Register configuration data

10.2

exception

MMU SVC OS

10.2.1

ARMv7-A ARMv7-R FIQ IRQ Supervisor

SVC

1

ARMv7-A IRQ FIQ

FIQ IRQ FIQ FIQ FIQ

FIQ IRQ FIQ IRQ

IRQ FIQ

2

ARMv7 MMU ARMv7

CPSR A A 1 CPU A

3

4

• Supervisor Call SVC

• Hypervisor HVC Hypervisor

• SMC

CPU

ARM

PL1 PL1 PL2

ARM Thumb CP15 SCTLR.TE ARM Thumb

10.2.2

Undef supervisor call SVC

ARM

IRQ FIQ FIQ

CPSR I IRQ F FIQ

SVC SVC

0x00 0x1C

0x0 0xFFFF0000

Security Extensions

Virtualization Extension



CPSR I F



10.2.3

ARM

0x00000000

ARM

0xFFFF0000 HIVECS

16 Thumb

B

PC 32MB

LDR PC [PC #offset]

| | | | |
|--------|----|----|---|
| offset | PC | 32 | B |
|--------|----|----|---|

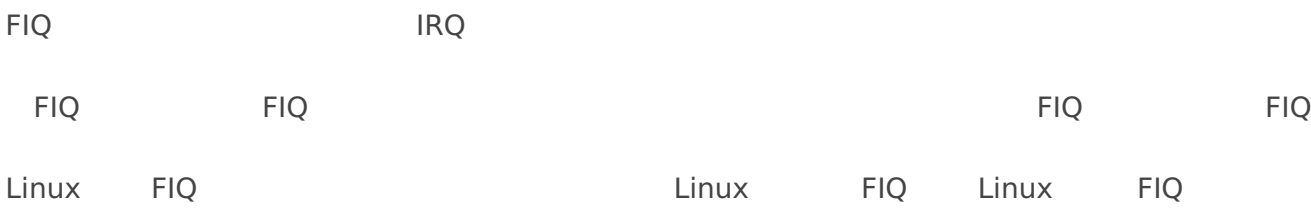
Hyp mode Hyp mode Hyp mode Hyp trap entry

8 8

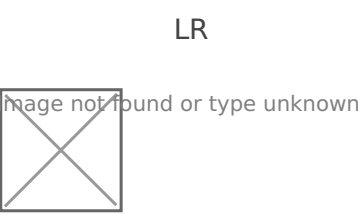
secure monitor MVBAR

secure PL1 monitor secure SCTRL.V V==0 secure VBAR

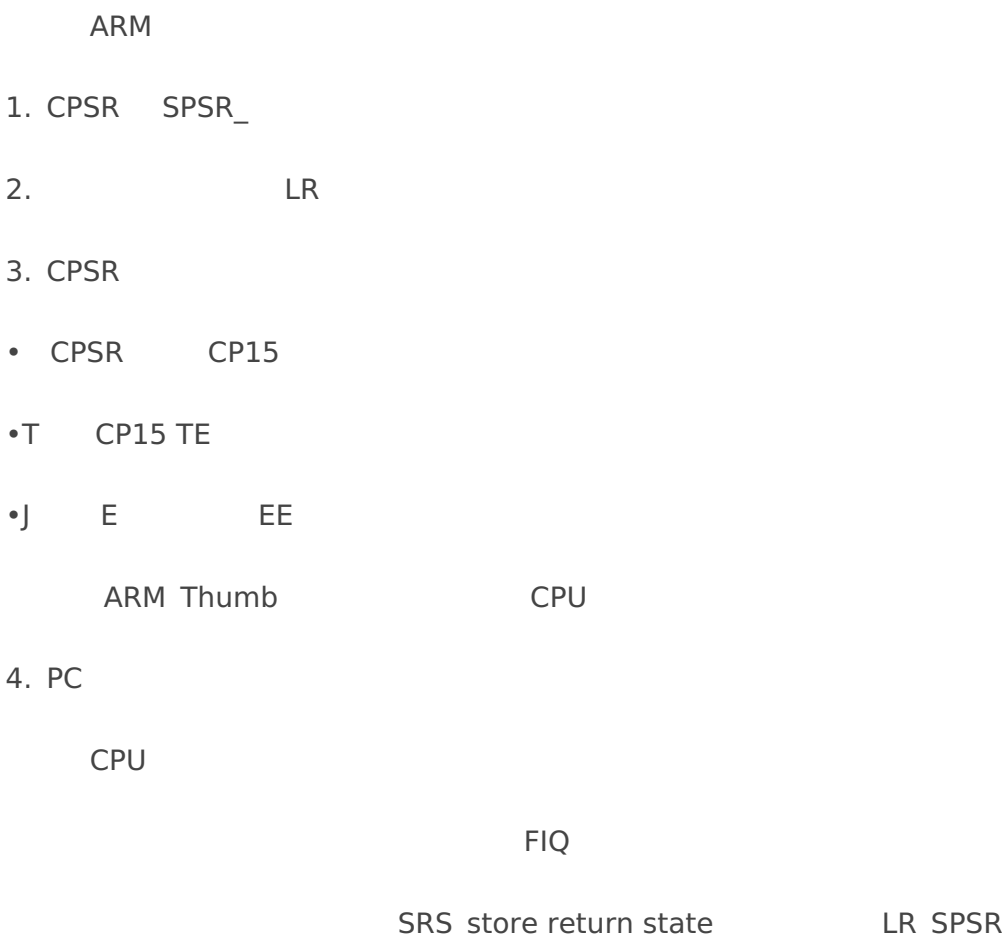
10.2.4 FIQ and IRQ



10.2.5



10.2.6



10.2.6.1

1. SPSR CPSR

2. PC

ARM RFE PC S SUBS PC LR #offset S RFE

• LR PC

```
SUBS PC, LR, 4
```

S SPSR CPSR

^

```
LDMFD sp!, {pc} ^  
  
LDMFD sp!, {R0-R12 pc} ^
```

^ SPSR CPSR

—

—

16 Thumb CPSR RFE PC SPSR

```
RFEFD sp
```

10.2.7

MMU

10.2.8

CPU

ARM

UNDEFINED

VFP

VFP

VFP

VFP

10.2.9 SVC

supervisor call SVC

OS

I / O

SVC

SVC

ARM Thumb

SVC

SPSR T

Thumb

ARM

ARM Thumb

SVC

Thumb

SVC

- LR-2 LR-4

- 16

- SVC 8 ARM 24

11-1 Linux SVC

SVC 0

ARM

SVC

R0

sys_write

sys_write

R0

10.3 und

10.3.1

** Git NoosProgramProject/(10_ /008_exception_undef)**

0xdeadC0de

printException

CPSR

Reset_Handler

SVC und

C

```
mcr p15, 0, r0, c12, c0, 0
```

Undefined_Handler

```
ldr pc, =Undefined_Handler
```

Undefined_Handler r0-r12 lr und

printException

CPSR

r0-r12

008_exception_undef\start.S

```
.text
.global _start, _vector_table
_start:
_vector_table:
[ldr [pc, =Reset_Handler]] /* Reset[] */
[ldr [pc, =Undefined_Handler]] /* Undefined instructions */
[/b Reset_Handler
[/b Undefined_Handler
[b halt[/b SVC_Handler[/ldr [pc, =SVC_Handler]] /* Supervisor Call [] */
[b halt[/ldr [pc, =PrefAbort_Handler]] /* Prefetch abort[] */
[b halt[/ldr [pc, =DataAbort_Handler]] /* Data abort[] */
[ word[0] /* RESERVED[] */
[b halt[/ldr [pc, =IRQ_Handler]] /* IRQ interrupt[] */
[b halt[/ldr [pc, =FIQ_Handler]] /* FIQ interrupt[] */

.align 2
Undefined_Handler:
[/*
[ * 1. lr_und
[ * 2. SPSR_und CPSR
[ * 3. CPSR M4-M0 11011, und
[ * 4. 0x4
[ */

[/* und r0-r12, */
[/* lr , */
[stmdb sp!, {r0-r12, lr}
```

```

/*      */
/* und */
mrs r0, cpsr
ldr r1, =und_string
bl printException

/*      */
ldmia sp!, {r0-r12, pc}^ /* ^ spsr cpsr */

und_string:
.string "undefined instruction exception"

.align 2
Reset_Handler:
/* Reset SCTLr Settings */
mrc p15, 0, r0, c1, c0, 0 /* read SCTRL, Read CP15 System Control register */
bic r0, r0, #(0x1 << 13) /* Clear V bit 13 to use normal exception vectors */
bic r0, r0, #(0x1 << 12) /* Clear I bit 12 to disable I Cache */
bic r0, r0, #(0x1 << 2) /* Clear C bit 2 to disable D Cache */
bic r0, r0, #(0x1 << 2) /* Clear A bit 1 to disable strict alignment */
bic r0, r0, #(0x1 << 11) /* Clear Z bit 11 to disable branch prediction */
bic r0, r0, #0x1 /* Clear M bit 0 to disable MMU */
mcr p15, 0, r0, c1, c0, 0 /* write SCTRL, Write to CP15 System Control register */

        cps      #0x1B                /* Enter undef mode */
        ldr       sp, =0x80300000      /* Set up undef mode stack */

        cps      #0x13                /* Enter Supervisor mode */
        ldr       sp, =0x80200000      /* Set up Supervisor Mode stack */
ldr r0, =_vector_table
mcr p15, 0, r0, c12, c0, 0 /* set VBAR, Vector Base Address Register */
/*mrc p15, 0, r0, c12, c0, 0 //read VBAR

bl clean_bss

bl system_init

und_code:

```

```

    word 0xdead0de /* undefine instruction */
    /.word 0xFFFFFFFF

    pl main

halt:
    b halt

clean_bss:
    /* BSS */
    ldr r1, =__bss_start
    ldr r2, =__bss_end
    mov r3, #0
clean:
    cmp r1, r2
    strlt r3, [r1]
    add r1, r1, #4
    plt clean

    mov pc, lr

```

10.3.2 4-1.4

10.3.3 4-1.4



image not found or type unknown

10.4 swi

10.4.1

Git **NoosProgramProject/(10_ /008_exception_swi)**

```
ldr pc, =SVC_Handler
```

| | | | | | | | | |
|-------------|--------|----|------|------|-----|-----------------------------|------|----|
| SVC_Handler | r0-r12 | lr | SVC | lr | R4 | printException | CPSR | R4 |
| r12 | lr | PC | SPSR | CPSR | swi | 008_exception_swi\start.S : | | |

```
.text
.global _start, _vector_table
_start:
_vector_table:
    ldr pc, =Reset_Handler    /* Reset          */
    ldr pc, =Undefined_Handler /* Undefined instructions */
    ldr pc, =SVC_Handler      /* Supervisor Call    */
    b halt//ldr pc, =PrefAbort_Handler /* Prefetch abort    */
    b halt//ldr pc, =DataAbort_Handler /* Data abort        */
    .word 0 /* RESERVED */
    b halt//ldr pc, =IRQ_Handler /* IRQ interrupt    */
    b halt//ldr pc, =FIQ_Handler /* FIQ interrupt    */

.align 2
Undefined_Handler:
/*      :
 * 1. lr_und
 * 2. SPSR_und      CPSR
 * 3. CPSR M4-M0    11011,    und
 * 4.    0x4
 */

/* und      r0-r12,    */
/* lr      ,    */
stmdb sp!, {r0-r12, lr}

/*      */
/* und */
mrs r0, cpsr
ldr r1, =und_string
bl printException

/*      */
```

```
    ldmia sp!, {r0-r12, pc}^ /* ^ spsr      cpsr */
```

```
und_string:
```

```
    string "undefined instruction exception"
```

```
.align 2
```

```
SVC_Handler:
```

```
    /*      :
```

```
    * 1. lr_svc
```

```
    * 2. SPSR_svc      CPSR
```

```
    * 3. CPSR M4-M0    10011,      svc
```

```
    * 4.      0x08
```

```
    */
```

```
    /*      */
```

```
    /* swi      r0-r12,      */
```

```
    /* lr      ,      */
```

```
    stmdb sp!, {r0-r12, lr}
```

```
    mov r4, lr
```

```
    /* swi      */
```

```
    mrs r0, cpsr
```

```
    ldr r1, =swi_string
```

```
    bl printException
```

```
    sub r0, r4, #4
```

```
    bl printSWIVal
```

```
    /*      */
```

```
    ldmia sp!, {r0-r12, pc}^ /* ^ spsr      cpsr */
```

```
swi_string:
```

```
    string "swi exception"
```

```
.align 2
```

```
Reset_Handler:
```

```
    /* Reset SCTlr Settings */
```

```
    mrc p15, 0, r0, c1, c0, 0 /* read SCTRL, Read CP15 System Control register */
```



```

bic r0, r0, #(0x1 << 13) /* Clear V bit 13 to use normal exception vectors */
bic r0, r0, #(0x1 << 12) /* Clear I bit 12 to disable I Cache*****/
bic r0, r0, #(0x1 << 2) /* Clear C bit 2 to disable D Cache*****/
bic r0, r0, #(0x1 << 2) /* Clear A bit 1 to disable strict alignment */
bic r0, r0, #(0x1 << 11) /* Clear Z bit 11 to disable branch prediction***/
bic r0, r0, #0x1000 /* Clear M bit 0 to disable MMU******/
mcr p15, 0, r0, c1, c0, 0 /* write SCTRL, Write to CP15 System Control register*/

cps #0x1B /* Enter undef mode */
ldr sp, =0x80300000 /* Set up undef mode stack */

cps #0x13 /* Enter Supervisor mode */
ldr sp, =0x80200000 /* Set up Supervisor Mode stack */

ldr r0, =_vector_table
mcr p15, 0, r0, c12, c0, 0 /* set VBAR, Vector Base Address Register*/
//mrc p15, 0, r0, c12, c0, 0 //read VBAR

bl clean_bss

bl system_init

und_code:
.word 0xdeadc0de /* undefine instruction */
//.word 0xFFFFFFFF

swi_code:
swi 0x123 /* , SWI , 0x8 */

bl main

halt:
b halt

clean_bss:
/* BSS */
ldr r1, =__bss_start
ldr r2, =__bss_end
mov r3, #0

```

clean:

cmp r1, r2

strlt r3, [r1]

add r1, r1, #4

bgt clean

mov pc, lr

10.4.2 4-1.4

10.4.3 4-1.4

Image not found or type unknown



10.5

ARM

v2 GIC

10.5.1

ARM

FIQ IRQ

FIQ IRQ

AI

CPSR

F I

CPS

CPSR A I F

IRQ FIQ

CPS IE CPS ID

A I F

Cortex-A

CPU

FIQ

FIQ

CPU

FIQ

10.5.2

ARM

GIC

10.5.3

image not found or type unknown



1. IRQ ARM PC LR_IRQ CPSR SPSR_IRQ CPSR IRQ
2. IRQ
- 3.
- 4.
5. SPSR_IRQ CPSR CPU LR_IRQ PC
FIQ

10.5.4

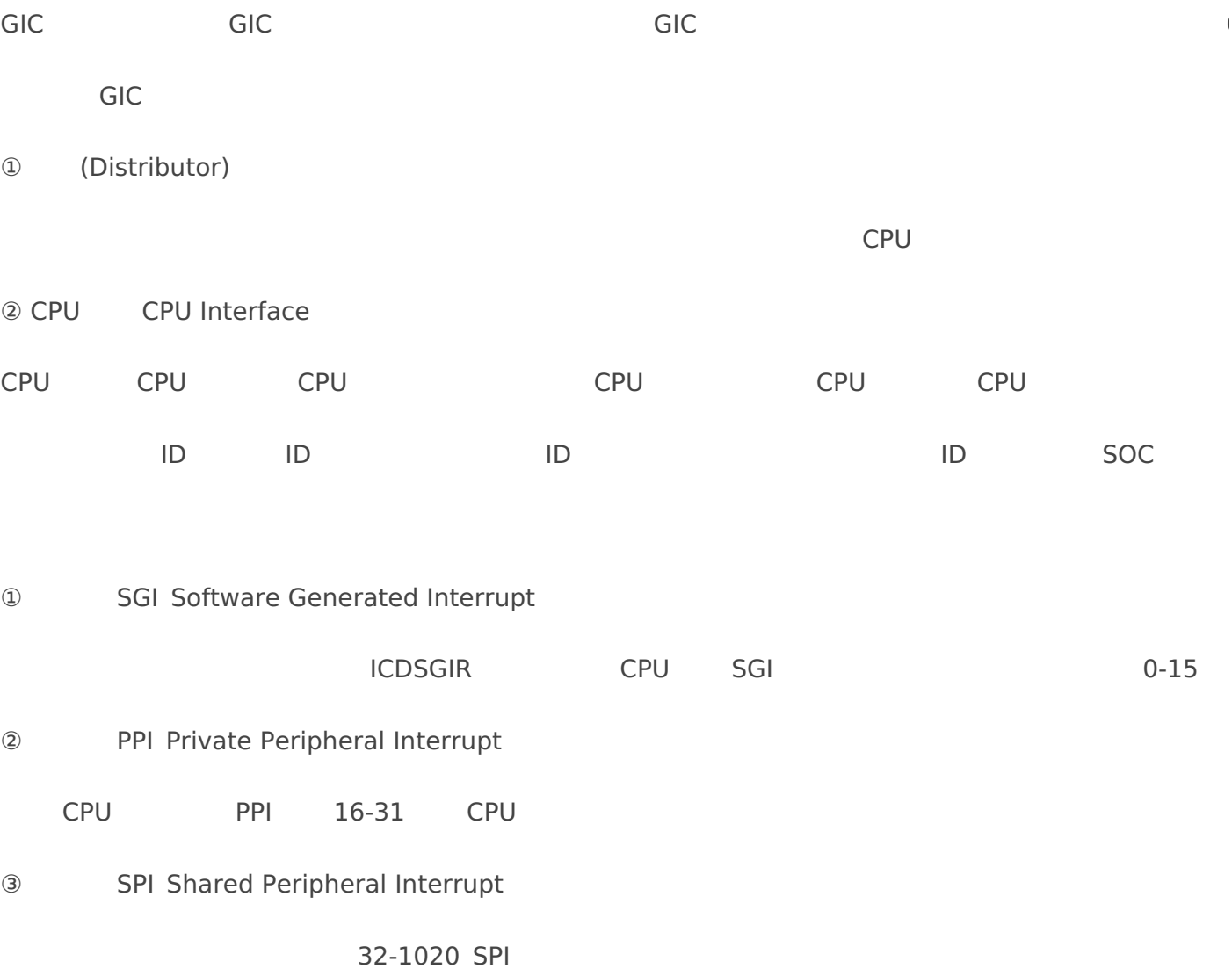
image not found or type unknown

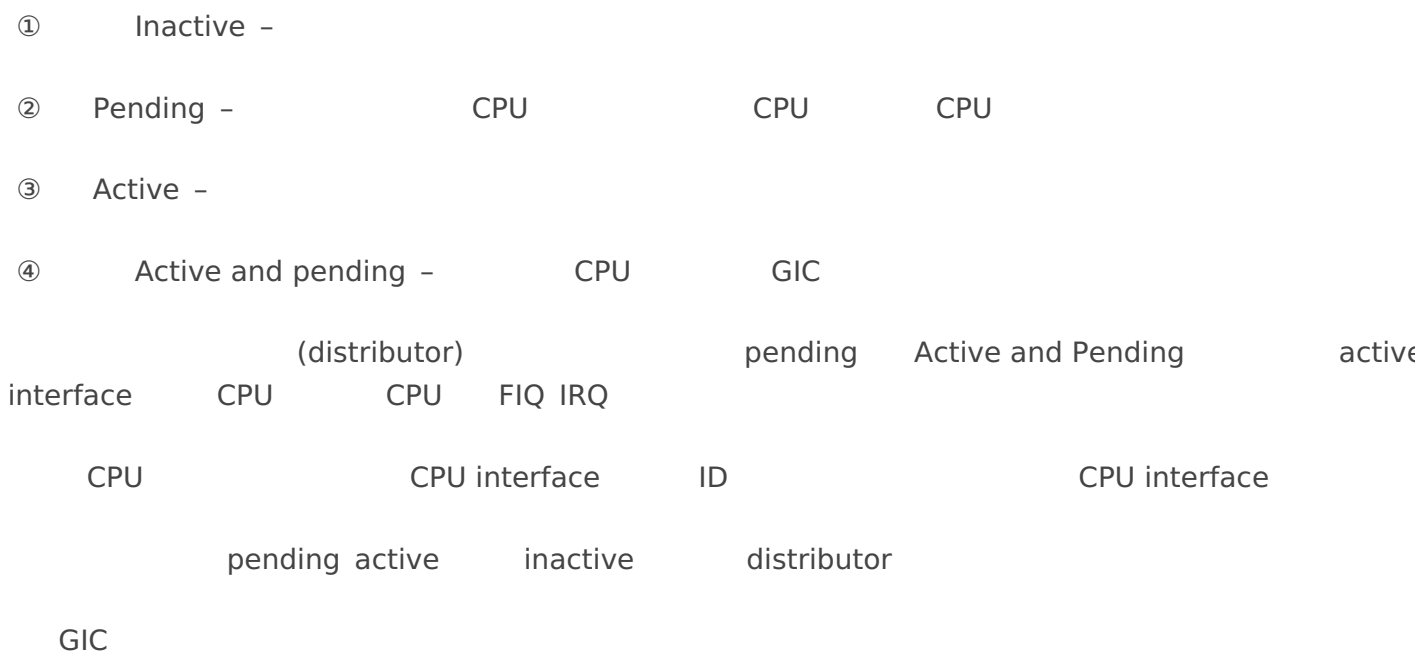


1. C IRQ CPU CPU
SPSR SPSR_irq SPSR
SRSFD sp !, 0x12
BL LR_IRQ BL Supervisor IRQ
2. SPSR_IRQ CPU

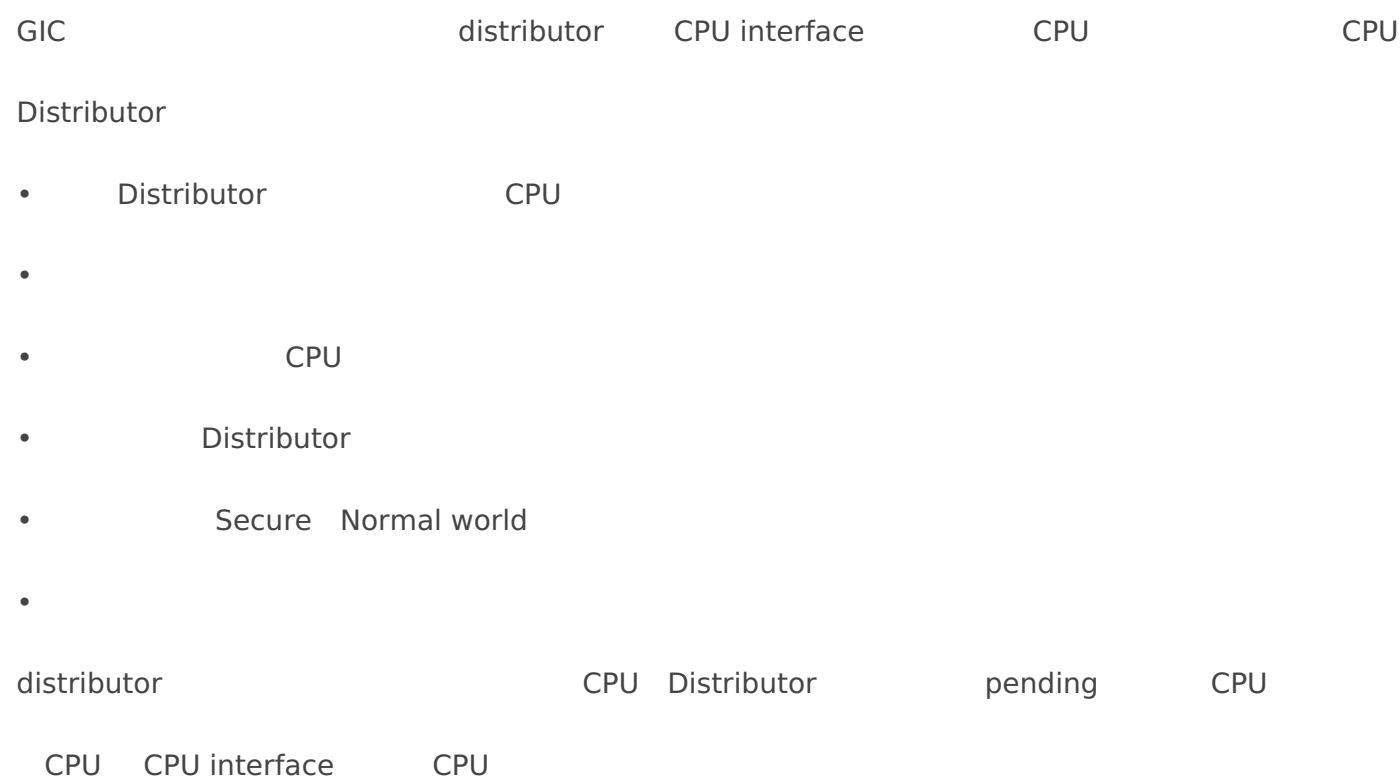
- 3. CPU SVC CPSR I 1
- 4.
- 5.
- 6. IRQ
- 7. PC CPSR CPSR SPSR I

10.6 GIC, Generic Interrupt Controller





10.6.1



10.6.2

The diagram illustrates the relationship between distributor and CPU components. It shows a distributor connected to a CPU via a distributor CPU interface. The GIC (Generic Interrupt Controller) is shown connected to the CPU. The CPSR (Current Program Status Register) is shown connected to the CPU. The diagram also shows a distributor connected to a CPU via a distributor CPU interface. The GIC is shown connected to the CPU. The CPSR is shown connected to the CPU. The diagram also shows a distributor connected to a CPU via a distributor CPU interface. The GIC is shown connected to the CPU. The CPSR is shown connected to the CPU.

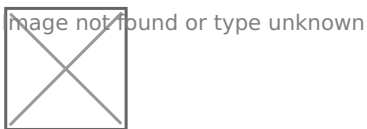
10.6.3 GIC

| | | | | |
|--------|---------------|---------------|--------------------------------|---------------|
| CPU | | CPU | Interrupt Acknowledge Register | ID |
| ID | distributor | active | ID | |
| | ID | CPU interface | End of Interrupt register | |
| active | inactive | pending | inactive and pending | CPU interface |
| | CPU interface | | ID 1023 | per |
| ID | | ID | CPU | |

10.7

10.7.1 Distributor

10.7.1.1 Distributor Control Register, GICD_CTLR



[1] EnableGrp1 pending Group 1 Distributor CPU interfaces

0 Group 1

1 Group 1

[0] EnableGrp0 pending Group 0 Distributor CPU interfaces

0 Group 0

1 Group 0

10.7.1.2 Interrupt Controller Type Register, GICD_TYPER

image not found or type unknown

[15:11] LSPI GIC SPI 0 0b00000 31 0b11111 0b00000 GIC

[10] SecurityExtn GIC

0

1

[7:5] CPUNumber CPU interfaces CPU interfaces 1 0b011 CPU int

[4:0] ITLinesNumber GIC ITLinesNumber = N

32*(N+1) ID 0 ID - 1

0b00011 128 ID 0-127

1020 0b11111 ID ID 1020-1023

10.7.1.3 Distributor Implementer Identification Register, GICD_IIDR

image not found or type unknown

[31:24] ProductID ID

[23:20]-

[19:16] Variant Variant

[15:12] Revision Revision

[11:0] GIC JEP106

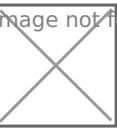
[11:8] JEP106 continuation code ARM 0x4

[7] 0

[6:0] JEP106code ARM [7:0] 0x3B

10.7.1.4 Interrupt Group Registers, GICD_IGROUPRn

image not found or type unknown



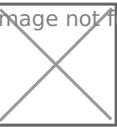
[31:0]

0 Group 0

1 Group 1

10.7.1.5 Interrupt Set-Enable Registers, GICD_ISENABLERn

image not found or type unknown



[31:0] SPI PPI Distributor CPU interfaces

0 1

0 1

10.7.1.6 Interrupt Clear-Enable Registers, GICD_ICENABLERn

image not found or type unknown



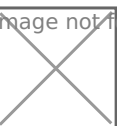
[31 0] SPI PPI Distributor CPU interfaces

0 1

0 1

10.7.1.7 Interrupt Set-Active Registers, GICD_ISACTIVERn

image not found or type unknown



[31:0] Set-active

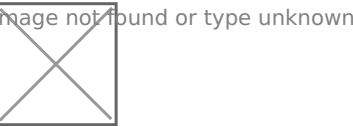
0 not active

1 active

0

1 Activates Activates Activates 1 1

10.7.1.8 Interrupt Clear-Active Registers, GICD_ICACTIVERn



[31:0] Clear-active

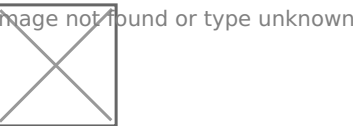
0 not active

1 active

0

1 active Deactivates Deactivates 1 0

10.7.1.9 Interrupt Priority Registers, GICD_IPRIORITYRn



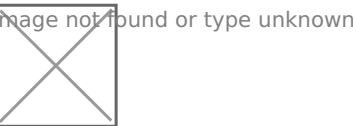
[31:24] byte offset 3

[23:16] byte offset 2

[15:8] byte offset 1

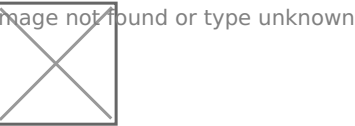
[7:0] byte offset 0

10.7.1.10 Interrupt Processor Targets Registers, GICD_ITARGETSRn



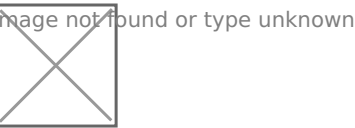
| | | | | | | |
|-------------|---------------|---|-----|-----|-----|----------|
| [31:24] CPU | byte offset 3 | 0 | CPU | 0x3 | 0 1 | GICD_IT/ |
| [23:16] CPU | byte offset 2 | | | | | |
| [15:8] CPU | byte offset 1 | | | | | |
| [7:0] CPU | byte offset 0 | | | | | |

10.7.1.11 Interrupt Configuration Registers, GICD_ICFGRn



| | | |
|--------------------------------|----------------|----------|
| [2F + 1:2F] Int_config field F | Int_config [1] | [2F + 1] |
| 0 | | |
| 1 | | |
| Int_config [0] | [2F] | |
| Sgi | | |
| Int_config [1] | RAO / WI | |
| PPI SPI | | |
| Int_config [1] SPI | PPI | |

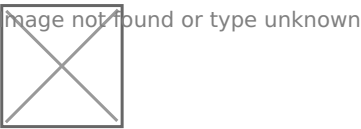
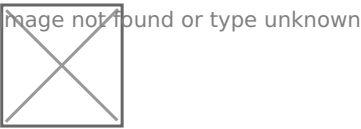
10.7.1.12 Identification registers



| | | | |
|-----------|--------------------|-----|---------|
| [31:8]- | CoreLink CoreSight | ID | RAZ ARM |
| [7:4] GIC | ArchRev | GIC | |
| •GICv1 | 0x1 | | |
| •GICv2 | 0x2 | | |
| [3:0]- | | | |

10.7.2 CPU interface

10.7.2.1 CPU Interface Control Register, GICC_CTLR



[9] EOImodeNS GICC_EOIR GICC_DIR

0 GICC_EOIR deactivate GICC_DIR

1 GICC_EOIR GICC_DIR deactivate

[6] IRQBypDisGrp1 CPU interface IRQ bypass IRQ

0 bypass IRQ

1 bypass IRQ

2-27 GICv2

[5] FIQBypDisGrp1 CPU interface FIQ bypass FIQ

0 FIQ

1 FIQ

2-27 GICv2

[0] EnableGrp1 CPU interface 1

0

1

0 CPU interface pending 1 1 CPU pending 1

10.7.2.2 Interrupt Priority Mask Register, GICC_PMR

image not found or type unknown



[7:0] CPU interface GIC 256 RAZ / WI

128 bit[0] = 0

64 bit [1 0] = 0b00

32 bit [2 0] = 0b0000

16 bit [3 0] = 0b00000

PS imx6ull 32

10.7.2.3 Binary Point Register, GICC_BPR

image not found or type unknown



[2 0] Binary point 8bit

GICC_CTLR.CBPR 1 GIC group 1

PS imx6ull BPR 2

10.7.2.4 Interrupt Acknowledge Register, GICC_IAR

image not found or type unknown



[12:10] CPUID SGI CPU interface 3 CPU interface 3

[9 0] ID ID

10.7.2.5 Interrupt Register, GICC_EOIR

image not found or type unknown



[12:10] CPUID SGI GICC_IAR CPUID SBZ

[9 0] EOINTID GICC_IAR ID