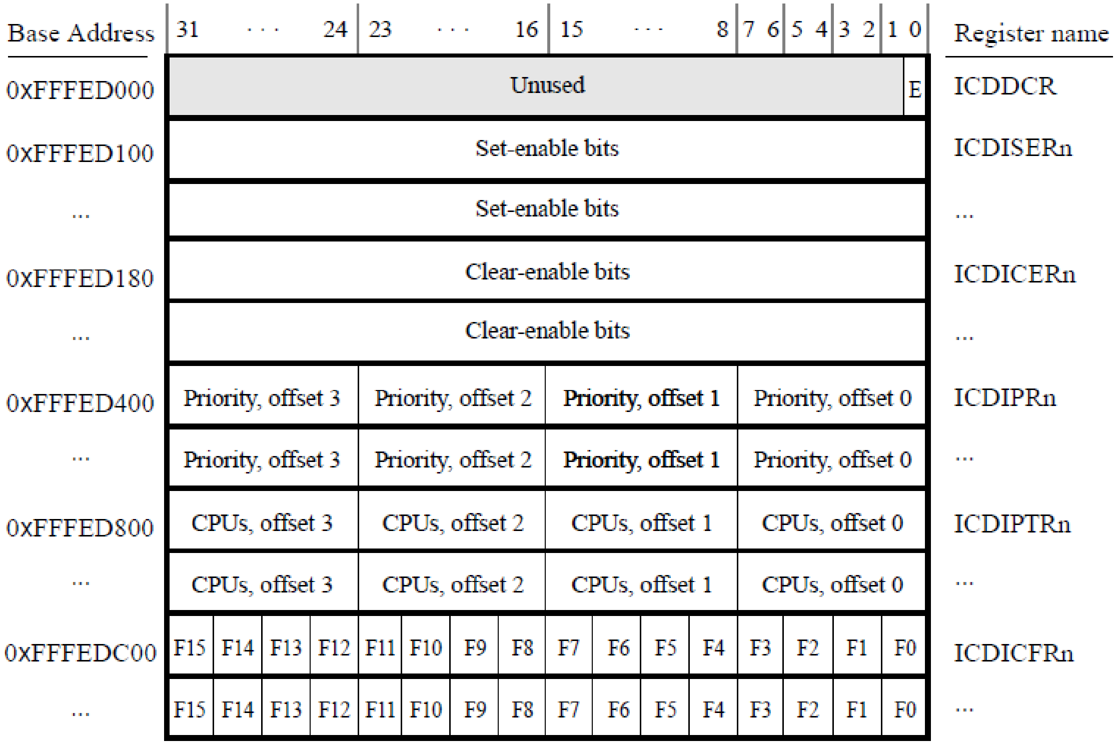
Example of Assembly Language Code

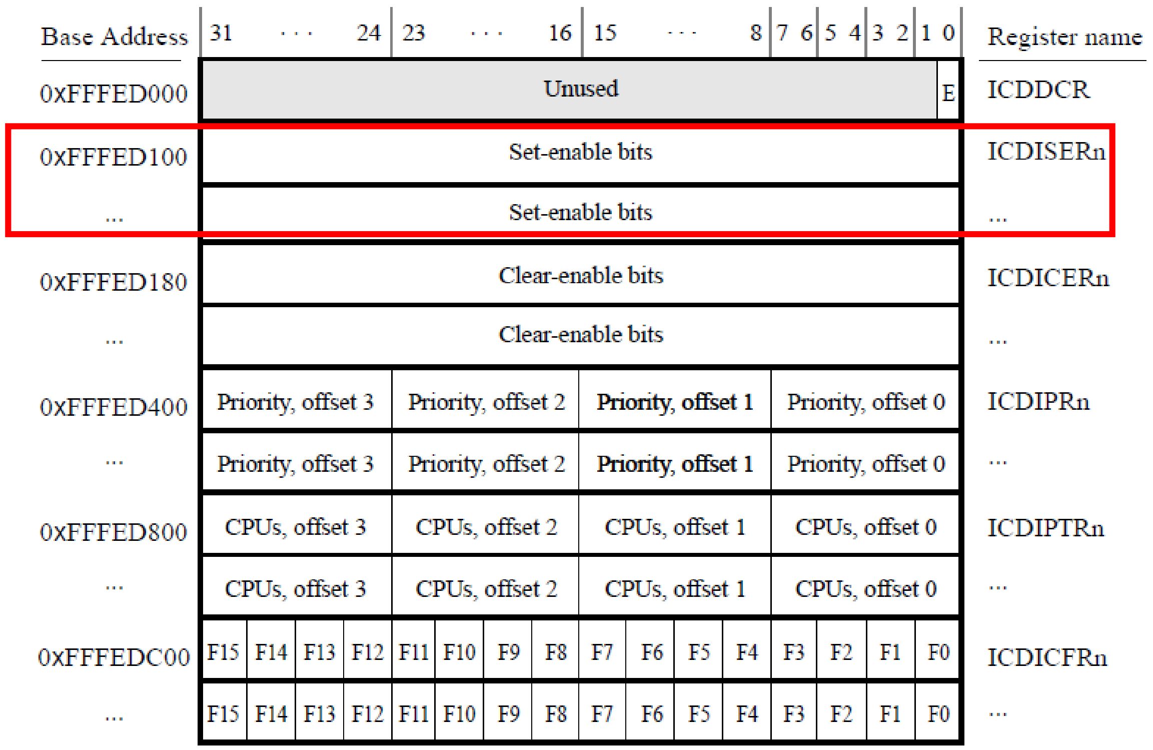
- Enable Interrupt ID **73** → parallel port connected to pushbutton KEYs in the DE1-SoC Computer



Base Address	31 . . . 24	23 . . . 16	15 . . . 8	7 6 5 4 3 2 1 0	Register name
0XFFFED000	Unused			E	ICDDCR
0XFFFED100	Set-enable bits				ICDISERn
...	Set-enable bits				...
0XFFFED180	Clear-enable bits				ICDICERn
...	Clear-enable bits				...
0XFFFED400	Priority, offset 3	Priority, offset 2	Priority, offset 1	Priority, offset 0	ICDIPRn
...	Priority, offset 3	Priority, offset 2	Priority, offset 1	Priority, offset 0	...
0XFFFED800	CPUs, offset 3	CPUs, offset 2	CPUs, offset 1	CPUs, offset 0	ICDIPTRn
...	CPUs, offset 3	CPUs, offset 2	CPUs, offset 1	CPUs, offset 0	...
0XFFFEDC00	F15 F14 F13 F12	F11 F10 F9 F8	F7 F6 F5 F4	F3 F2 F1 F0	ICDICFRn
...	F15 F14 F13 F12	F11 F10 F9 F8	F7 F6 F5 F4	F3 F2 F1 F0	...

Dist. Reg.

Interrupt ID **73** a. ICDISERn



Base Address	31 . . . 24	23 . . . 16	15 . . . 8	7 6 5 4 3 2 1 0	Register name
0xFFFED000	Unused			E	ICDDCR
0xFFFED100	Set-enable bits				ICDISERn
...	Set-enable bits				...
0xFFFED180	Clear-enable bits				ICDICERn
...	Clear-enable bits				...
0xFFFED400	Priority, offset 3	Priority, offset 2	Priority, offset 1	Priority, offset 0	ICDIPRn
...	Priority, offset 3	Priority, offset 2	Priority, offset 1	Priority, offset 0	...
0xFFFED800	CPUs, offset 3	CPUs, offset 2	CPUs, offset 1	CPUs, offset 0	ICDIPTRn
...	CPUs, offset 3	CPUs, offset 2	CPUs, offset 1	CPUs, offset 0	...
0xFFFEDC00	F15 F14 F13 F12	F11 F10 F9 F8	F7 F6 F5 F4	F3 F2 F1 F0	ICDICFRn
...	F15 F14 F13 F12	F11 F10 F9 F8	F7 F6 F5 F4	F3 F2 F1 F0	...

Dist. Reg.

- The Interrupt Set Enable Registers (ICDISERn) are used to **enable the forwarding of each supported interrupt** from the Distributor to the CPU Interface.
- The **n** postfix in the name ICDISERn means that multiple registers exist.
- The set-enable bits for the first 32 Interrupt IDs are provided in the register at address 0xFFFED100, the next 32 are provided in the register at the following word address, which is 0xFFFED104, and so on.
- Given a specific Interrupt ID, *N*, the address of the register that contains its set-enable bit is given by the integer calculation

$$\text{address} = \text{0xFFFED100} + (N / 32)*4,$$

and the index of the bit inside this register is given by

$$\text{index} = N \bmod 32.$$

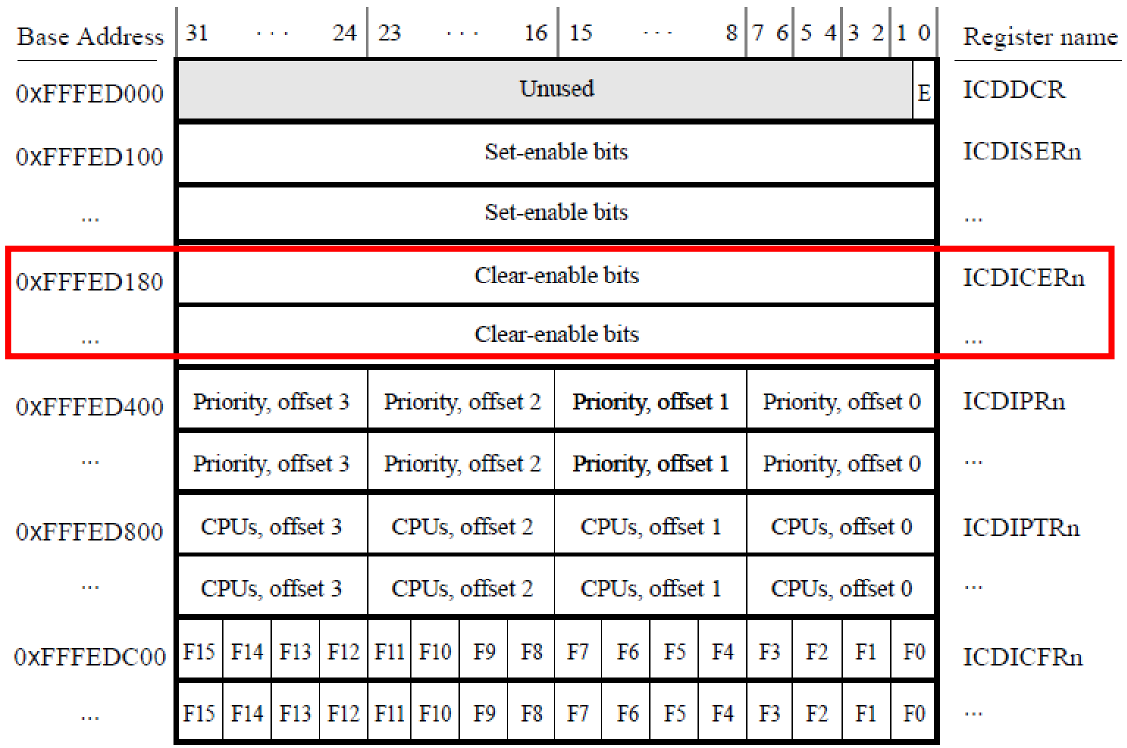
Writing the value 1 into a set-enable bit enables the forwarding of the corresponding IRQ to the CPU Interface.

- Example: N = 73 →

address = 0xFFFED100 + (3*4) = **0xFFFED10C**

Index = 73 mod 32 = **9**

Interrupt ID **73 b. ICDICERn**

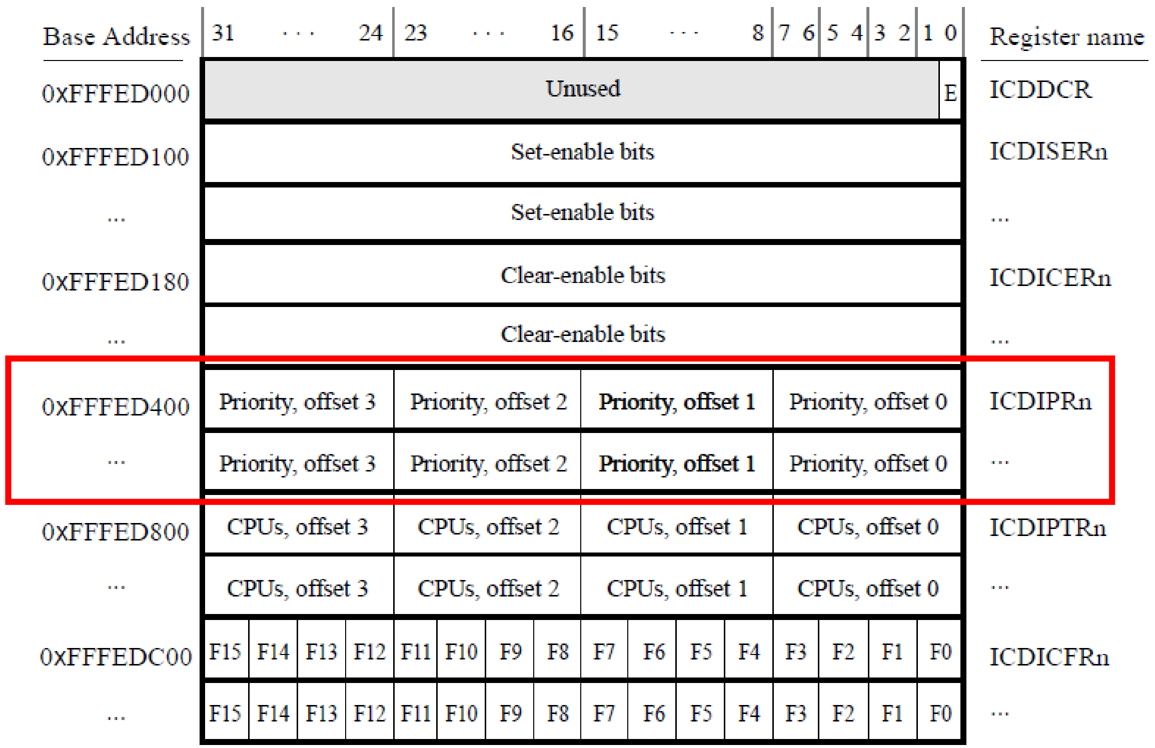


Base Address	31 . . . 24	23 . . . 16	15 . . . 8	7 6 5 4 3 2 1 0	Register name
0xFFFED000	Unused			E	ICDDCR
0xFFFED100	Set-enable bits				ICDISERn
...	Set-enable bits				...
0xFFFED180	Clear-enable bits				ICDICERn
...	Clear-enable bits				...
0xFFFED400	Priority, offset 3	Priority, offset 2	Priority, offset 1	Priority, offset 0	ICDIPRn
...	Priority, offset 3	Priority, offset 2	Priority, offset 1	Priority, offset 0	...
0xFFFED800	CPUs, offset 3	CPUs, offset 2	CPUs, offset 1	CPUs, offset 0	ICDIPTRn
...	CPUs, offset 3	CPUs, offset 2	CPUs, offset 1	CPUs, offset 0	...
0xFFFEDC00	F15 F14 F13 F12	F11 F10 F9 F8	F7 F6 F5 F4	F3 F2 F1 F0	ICDICFRn
...	F15 F14 F13 F12	F11 F10 F9 F8	F7 F6 F5 F4	F3 F2 F1 F0	...

Dist. Reg.

- In the same way that each supported interrupt can be enabled by using ICDISERn, each interrupt can be disabled by using the Interrupt Clear Enable Registers (ICDICERn). The method for calculating the address and index for ICDICERn is the same as that for ICDISERn, except that the base address is 0xFFFED180.

Interrupt ID **73** c. ICDIPRn



Base Address	31 . . . 24	23 . . . 16	15 . . . 8	7 6 5 4 3 2 1 0	Register name
0xFFFED000	Unused			E	ICDDCR
0xFFFED100	Set-enable bits				ICDISERn
...	Set-enable bits				...
0xFFFED180	Clear-enable bits				ICDICERn
...	Clear-enable bits				...
0xFFFED400	Priority, offset 3	Priority, offset 2	Priority, offset 1	Priority, offset 0	ICDIPRn
...	Priority, offset 3	Priority, offset 2	Priority, offset 1	Priority, offset 0	...
0xFFFED800	CPUs, offset 3	CPUs, offset 2	CPUs, offset 1	CPUs, offset 0	ICDIPTRn
...	CPUs, offset 3	CPUs, offset 2	CPUs, offset 1	CPUs, offset 0	...
0xFFFEDC00	F15 F14 F13 F12	F11 F10 F9 F8	F7 F6 F5 F4	F3 F2 F1 F0	ICDICFRn
...	F15 F14 F13 F12	F11 F10 F9 F8	F7 F6 F5 F4	F3 F2 F1 F0	...

Dist. Reg.

- The Interrupt Priority Registers (ICDIPRn) are used to associate a priority level with each individual interrupt. On reset, these registers are set to 0x00000000, which represents the highest priority. Each Interrupt ID's priority field is one byte in size, which means that the register at the base address holds the priority levels for Interrupt IDs from 0 to 3. The priority levels for the next four Interrupt IDs use the register at address 0xFFFED404, and so on.
- Setting the priority field for an Interrupt ID to a larger number results in lower priority for the corresponding interrupt.

address = 0xFFFED400 + (*N* / 4)*4

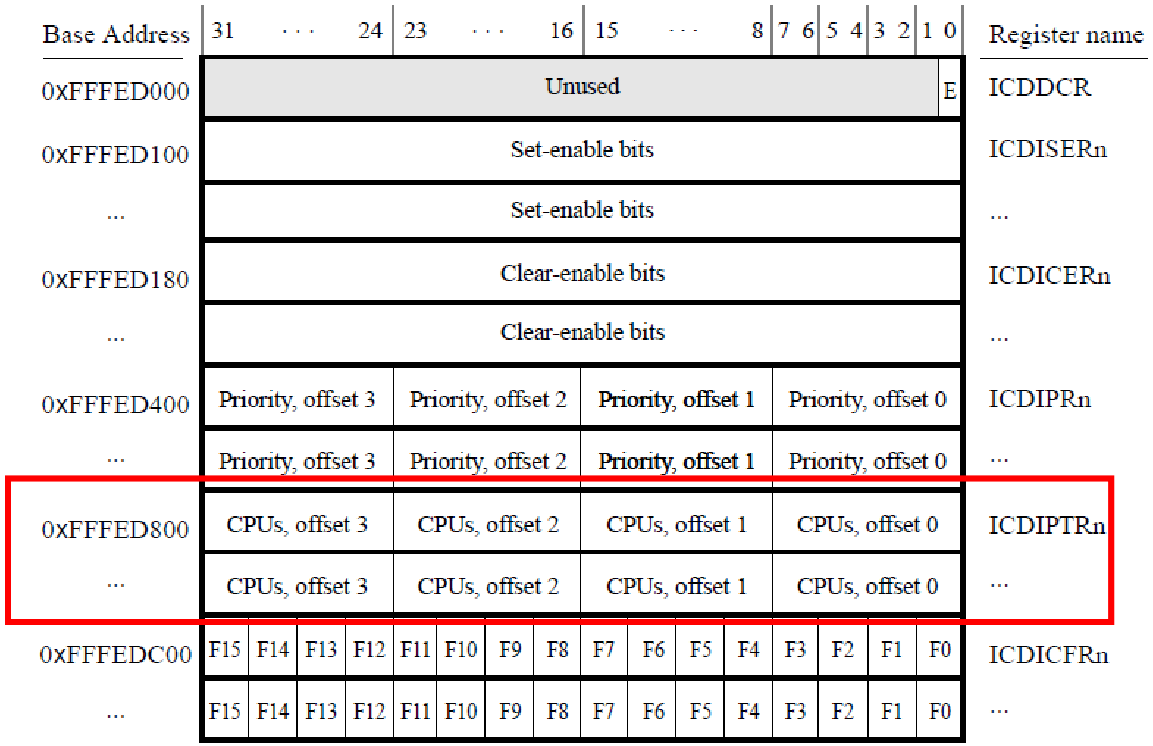
offset = *N* mod 4.

- Example: N = 73 →

address = 0xFFFED400 + (18*4) = **0xFFFED448**

Offset = 73 mod 4 = **1**

Interrupt ID **73 d. ICDIPTRn**



Base Address	31 · · · 24	23 · · · 16	15 · · · 8	7 6 5 4 3 2 1 0	Register name
0XFFFED000	Unused			E	ICDDCR
0XFFFED100	Set-enable bits				ICDISERn
...	Set-enable bits				...
0XFFFED180	Clear-enable bits				ICDICERn
...	Clear-enable bits				...
0XFFFED400	Priority, offset 3	Priority, offset 2	Priority, offset 1	Priority, offset 0	ICDIPRn
...	Priority, offset 3	Priority, offset 2	Priority, offset 1	Priority, offset 0	...
0XFFFED800	CPUs, offset 3	CPUs, offset 2	CPUs, offset 1	CPUs, offset 0	ICDIPTRn
...	CPUs, offset 3	CPUs, offset 2	CPUs, offset 1	CPUs, offset 0	...
0XFFFEDC00	F15 F14 F13 F12	F11 F10 F9 F8	F7 F6 F5 F4	F3 F2 F1 F0	ICDICFRn
...	F15 F14 F13 F12	F11 F10 F9 F8	F7 F6 F5 F4	F3 F2 F1 F0	...

Dist. Reg.

- The Interrupt Processor Targets Registers (ICDIPTRn) are used to specify the CPU interfaces to which each interrupt. the CPUs field for each Interrupt ID is one byte in size. This size is used because some versions of the ARM A9 MPCORE have up to eight A9 cores. A target CPU is selected by setting its corresponding bit field to 1. Thus, setting the byte at address 0xFFFED800 to the value 0x01 would target Interrupt ID 0 to CPU 0, setting this same byte to 0x02 would target CPU 1, and setting the byte to the value 0x03 would target both CPU 0 and CPU 1. The scheme for calculating the address of the ICDIPTRn register for a specific Interrupt ID, and also its byte index, is the same as the one shown above for ICDIPRn.

$$\text{address} = \text{0xFFFED800} + (N / 4)*4$$

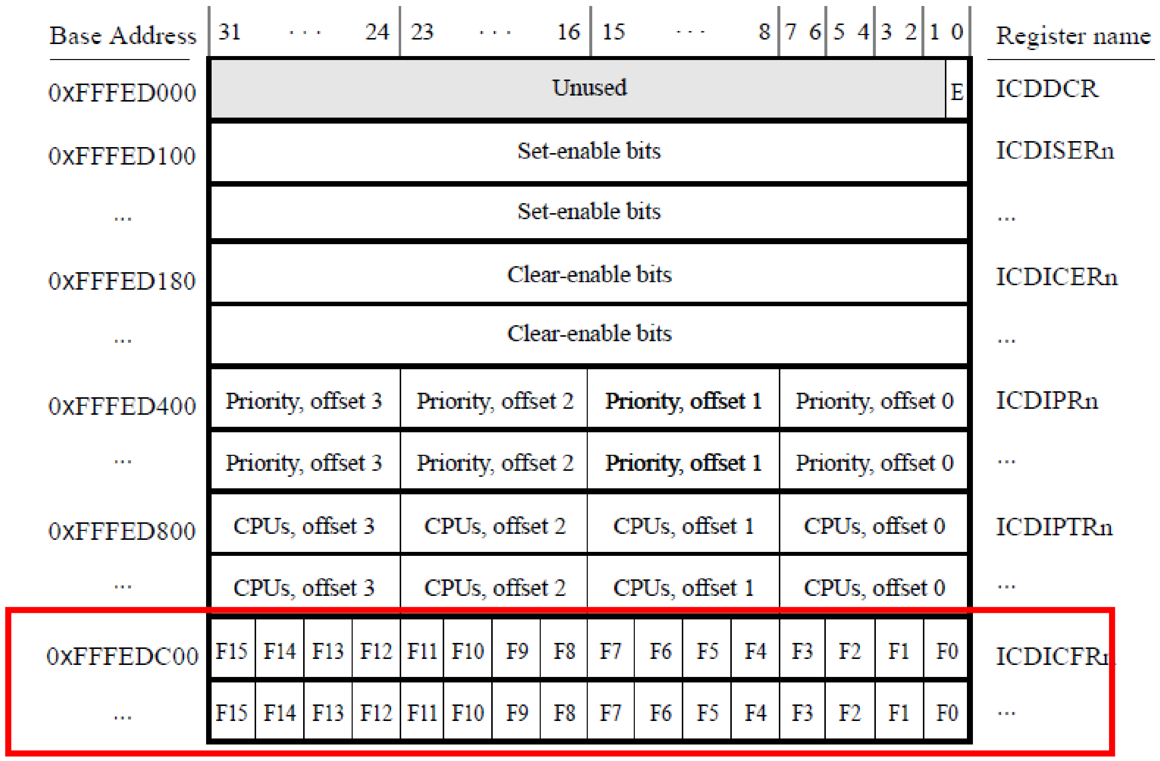
$$\text{offset} = N \bmod 4.$$

- Example: N = 73 →

address = 0xFFFED800 + (18*4) = **0xFFFED848**

Offset = 73 mod 4 = **1**

Interrupt ID **73** e. ICDICFRn



Base Address	31 · · · 24				23 · · · 16				15 · · · 8				7 6	5 4	3 2	1 0	Register name
0xFFFED000	Unused															E	ICDDCR
0xFFFED100	Set-enable bits																ICDISERn
...	Set-enable bits																...
0xFFFED180	Clear-enable bits																ICDICERn
...	Clear-enable bits																...
0xFFFED400	Priority, offset 3				Priority, offset 2				**Priority, offset 1**				Priority, offset 0				ICDIPRn
...	Priority, offset 3				Priority, offset 2				**Priority, offset 1**				Priority, offset 0				...
0xFFFED800	CPUs, offset 3				CPUs, offset 2				CPUs, offset 1				CPUs, offset 0				ICDIPTRn
...	CPUs, offset 3				CPUs, offset 2				CPUs, offset 1				CPUs, offset 0				...
0xFFFEDC00	F15	F14	F13	F12	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1	F0	ICDICFRn
...	F15	F14	F13	F12	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1	F0	...

Dist. Reg.

- The Interrupt Configuration Registers (ICDICFRn) are used to specify whether each supported interrupt should be handled as level- or edge-sensitive by the GIC. There is a **two-bit** field associated with each Interrupt ID. The least-significant bit in this field is not used. Setting the most-significant bit of this field to 1 makes the corresponding interrupt signal edge-sensitive, and setting this field to 0 makes it level-sensitive.
- The first 16 Interrupt IDs use the ICDICFRn register at address 0xFFFEDC00, the next 16 at address 0xFFFEDC04, and so on. Given a specific Interrupt ID, *N*, the address of the ICDICFRn

address = 0xFFFEDC00 + (*N* / 16)*4

and the index of the bit inside this register is given by

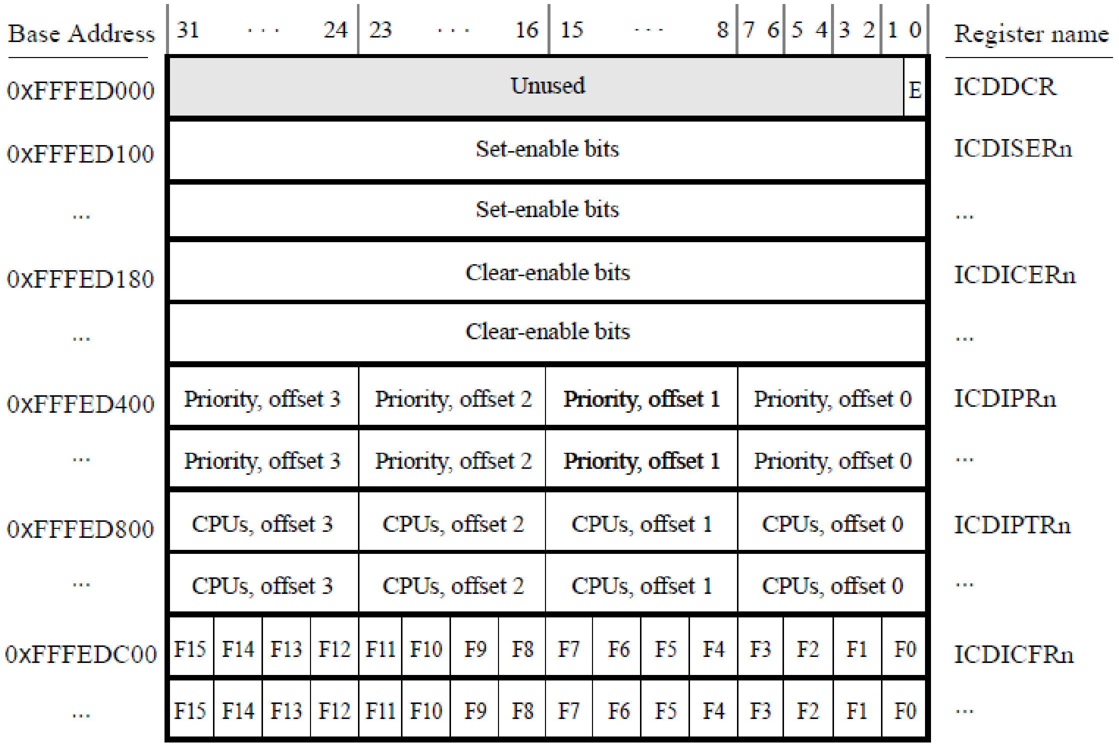
index = 2*(*N* mod 16) +1.

- Example: N = 73 →

address = 0xFFFEDC00 + (4*4) = **0xFFFEDC10**

Index = 73 mod 16 +1 = **19**

Interrupt ID **73 summary**



Base Address	31 . . . 24	23 . . . 16	15 . . . 8	7 6 5 4 3 2 1 0	Register name
0xFFFED000	Unused			E	ICDDCR
0xFFFED100	Set-enable bits				ICDISERn
...	Set-enable bits				...
0xFFFED180	Clear-enable bits				ICDICERn
...	Clear-enable bits				...
0xFFFED400	Priority, offset 3	Priority, offset 2	**Priority, offset 1**	Priority, offset 0	ICDIPRn
...	Priority, offset 3	Priority, offset 2	**Priority, offset 1**	Priority, offset 0	...
0xFFFED800	CPUs, offset 3	CPUs, offset 2	CPUs, offset 1	CPUs, offset 0	ICDIPTRn
...	CPUs, offset 3	CPUs, offset 2	CPUs, offset 1	CPUs, offset 0	...
0xFFFEDC00	F15 F14 F13 F12	F11 F10 F9 F8	F7 F6 F5 F4	F3 F2 F1 F0	ICDICFRn
...	F15 F14 F13 F12	F11 F10 F9 F8	F7 F6 F5 F4	F3 F2 F1 F0	...

Dist. Reg.

- N = 73 → Note that following operation clears all other interrupts.

ICDISER: address = **0xFFFED10C,** Index = **9**

LDR R0,=**0xFFFED10C**

MOV R1,#1

LSL R1, #9

STR R1,[R0]

ICDIPTR: address = **0xFFFED848,** Offset = **1**

LDR R0,=**0xFFFED848**

MOV R1,#0x04 // assign a CPU

STR R1,[R0]

- N = 73 → Note that following operation clears all other interrupts.

IICDIPR: address = **0xFFFED448,** Offset = **1**

LDR R0,=**0xFFFED448**

MOV R1,#0xFF // priority is 1 byte.

LSL R1, #8 // shift by one byte.

STR R1,[R0]

ICDICFR: address = **0xFFFEDC10,** Index = **19**

LDR R0,=**0xFFFED10C**

MOV R1,#1

LSL R1, #19

STR R1,[R0]

Or, Leave defaults.