

Must enable interrupts/configure them in all three places

- ① In the processor, assume the processor starts in supervisor mode → has the privilege to change the CPSR.

`MSR CPSR.C, #0b01010011`

• 1 = 0 means supervisor mode

I=1: Interrupts are disabled

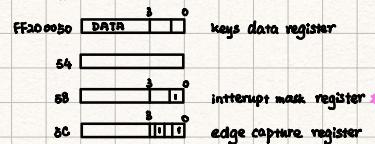
I=0 " " " enabled (respond to IRQ as above)

\* To enable interrupts, use the MSR instruction

`MSR CPSR.C, #0b01010011`

- ② Enable the KEYS to cause an interrupt

Address



\* Turn on the bit in the interrupt mask register to enable the corresponding bit in edge capture register interrupt to GIC

③ Your interrupt service routine must:
 

- Read the edge capture register to determine which reg is pressed - should respond to I
- When done the routine must reset that bit in the edge capture reg. As in labs

- ④ Enable/configure interrupts in the GIC → too complicated, just use `BL CONFIG-GIC`

but you do need two memory mapped reg in the GIC to deal with multiple interrupts

Address



- Each possible source, from a device, of an interrupt, has its own numerical 10-bit ID number.

Bits 9-0 of ICCIAR tell you the ID of the device causing the interrupt.

KEYS ID = 7310

A9\_PRIVATE\_TIMER = 2910

Interval Timer = 7210

Bits 9-0 of ICCEDIR should be written with the ID of devices that you want to acknowledge & have GIC turn off IRQ

Steps Required in interrupts:

- ① Set up the Exception Vector Table - vector @ D018

- ② Initialize the stack pointers

- for both supervisor mode & IRQ mode

`MSR CPSR.C, #0b11010011` // disabled interrupt

`LDR SP, =0x20000` // user's IRQ banked SP

`MSR CPSR.C, #0b11010011` // go back to supervisor mode

`LDR SP, =0x40000` // set supervisor stack pointer

different reg

- ③ Configure GIC + BL CONFIG-GIC

- ④ Enable interrupts in the device

e.g. the keys as shown above

- ⑤ Turn on processor interrupts → bit 1 increases → 0

- ⑥ Write the IRQ handler that figures out which device is causing interrupt & call the appropriate service routine

\* make sure to turn off GIC device interrupt for each instance of an interrupt.