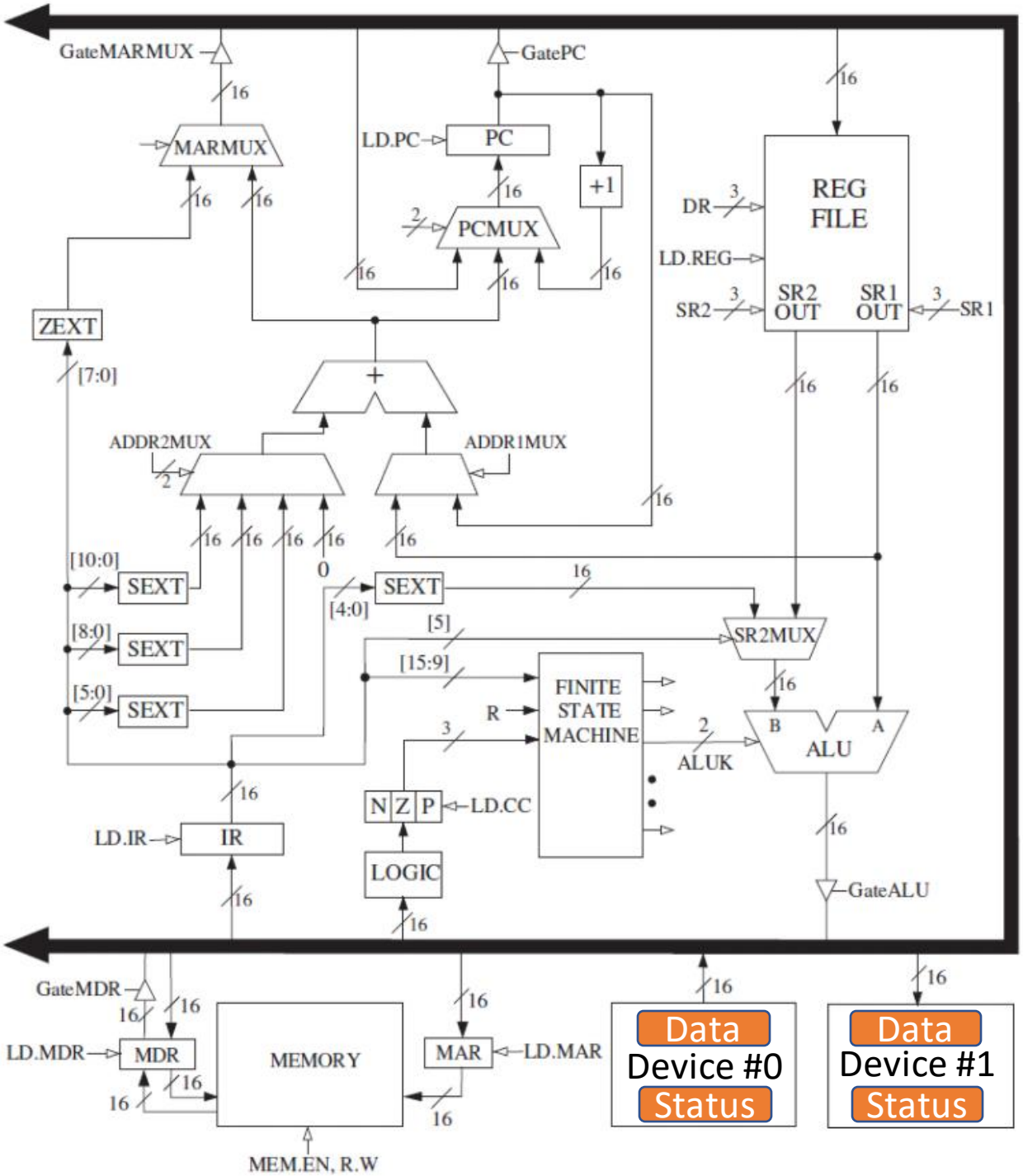
TRAP (System Call)

EECS388 Fall 2022



Context

Recommended reading: Chapter 9 of "Introduction to Computing," Patt, Patel

Memory Mapped I/O

```
        .ORIG x3000
        LDI   R5, DDR0
DDR0    .FILL xFE04
```

What are the issues with letting a user directly access device registers?

What are the issues with letting a user directly access device registers?

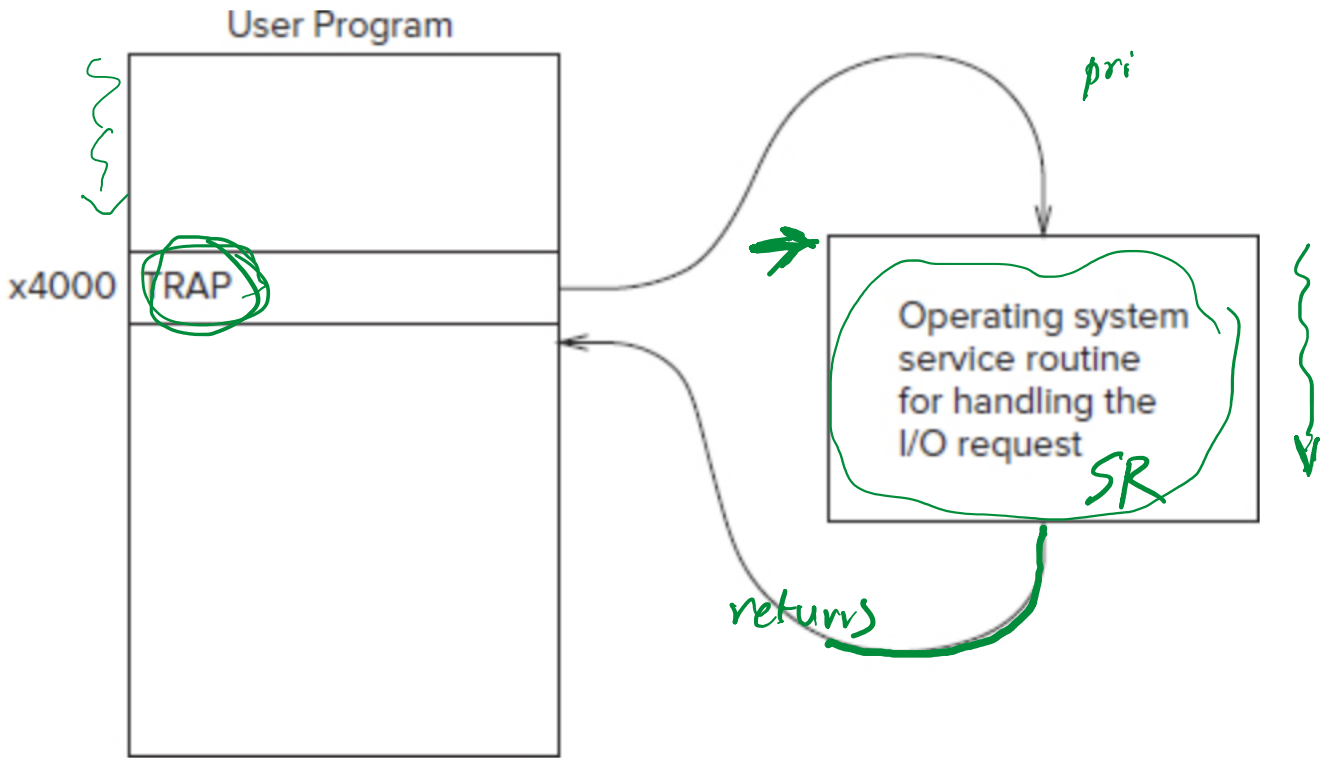
- Programmers need to know about low-level device interactions

- Device registers are shared => security and safety issues

Solution: Using TRAP Instruction (System Call)

- TRAP: request service from OS running in *privilege mode*
- Recall that a process running in *privilege mode* could execute any instruction

- User execute TRAP with a specific trapvector
  - Everything else happens under the hood



The TRAP Mechanism

- *Service Routine*
- *Trap Vector Table*
- *The TRAP Instruction*
- *A Linkage*

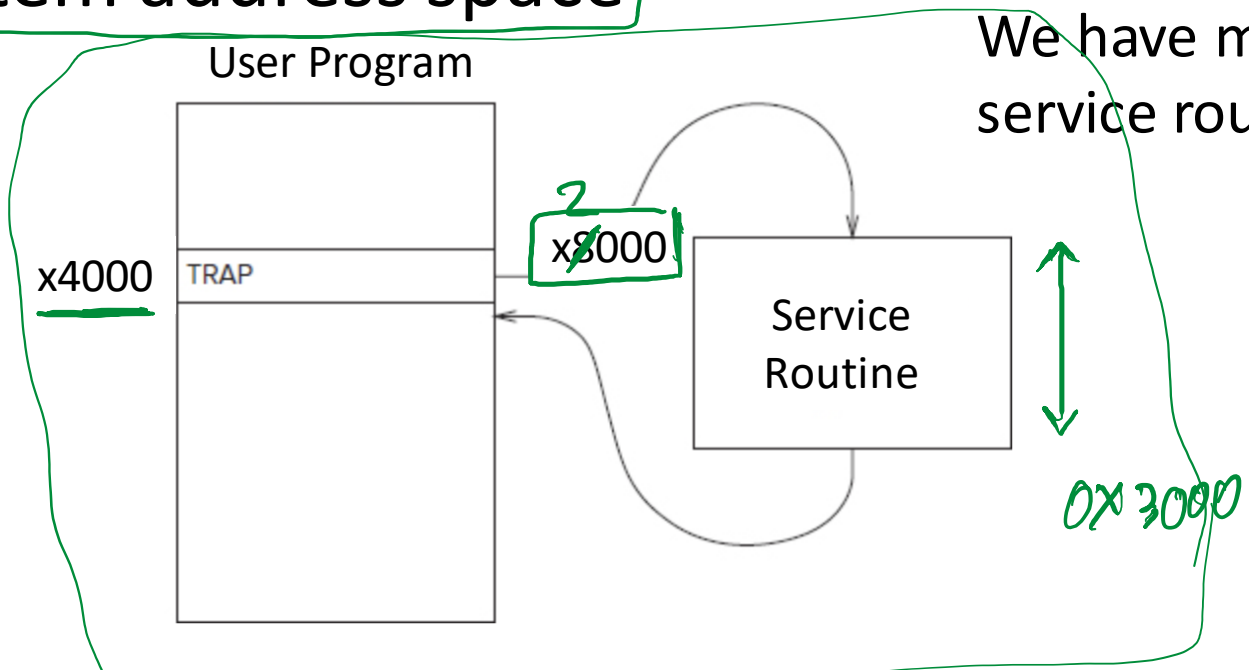
The TRAP Mechanism

- *Service Routine*
- *Trap Vector Table*
- *The TRAP Instruction*
- *A Linkage*

Service Routine

- A function that execute on behalf of user by the OS
  - Placed in arbitrary addresses in the system address space

We have multiple service routines



The TRAP Mechanism

- *Service Routine*
- *Trap Vector Table*
- *The TRAP Instruction*
- *A Linkage*

Trap Vector Table

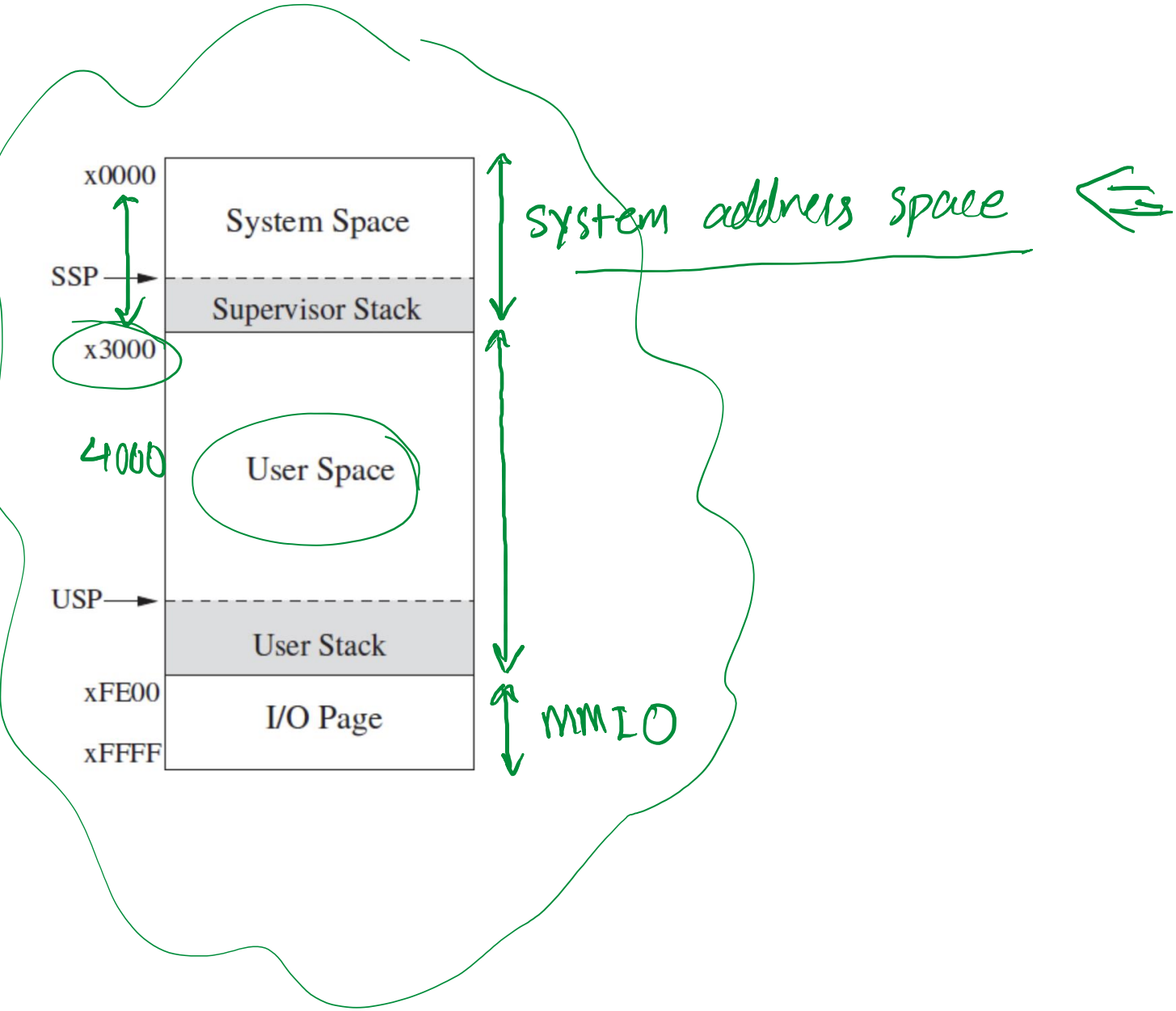
TRAP 0x20

- A table of the starting addresses of service routines

Start address of a subroutine that:

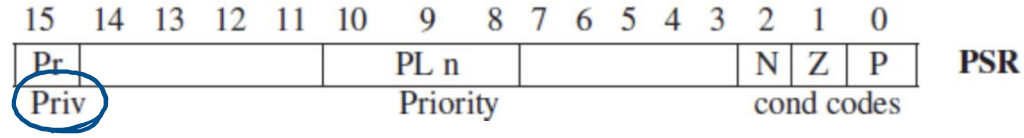
1- reads a single char from keyboard
2- writes a character into console display
3- writes a string into console display
4- reads a single char and echoes it to the console display
5- writes a string into console display
6- halts the computer

x0000	⋮	
x0020	x03E0	1
x0021	x0420	2
x0022	x0460	3
x0023	x04A0	4
x0024	x04E0	5
x0025	x0520	6
⋮	⋮	
x00FF		


x0000
System Space
SSP
Supervisor Stack
x3000
4000
User Space
USP
User Stack
xFE00
I/O Page
xFFFF
system address space
MMIO

The TRAP Mechanism

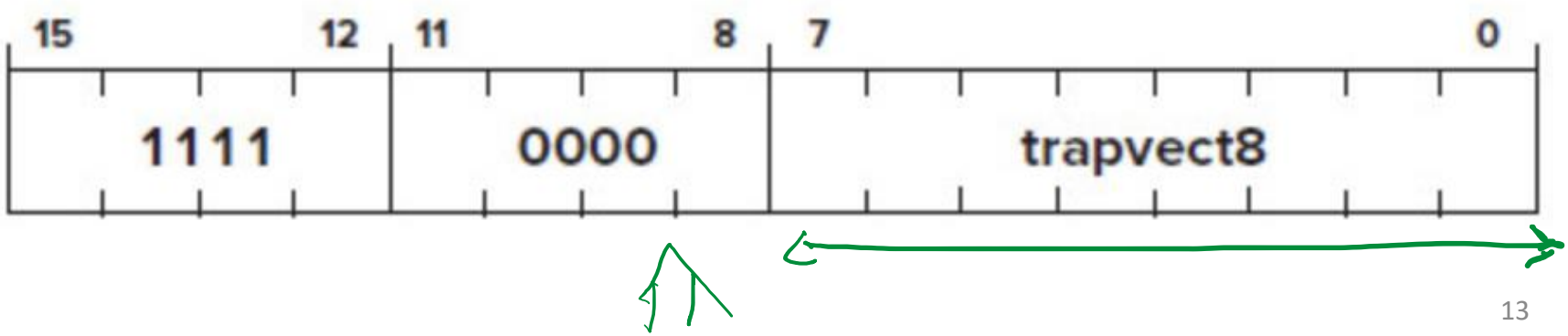
- *Service Routine*
- *Trap Vector Table*
- *The TRAP Instruction*
- *A Linkage*



TRAP Instruction

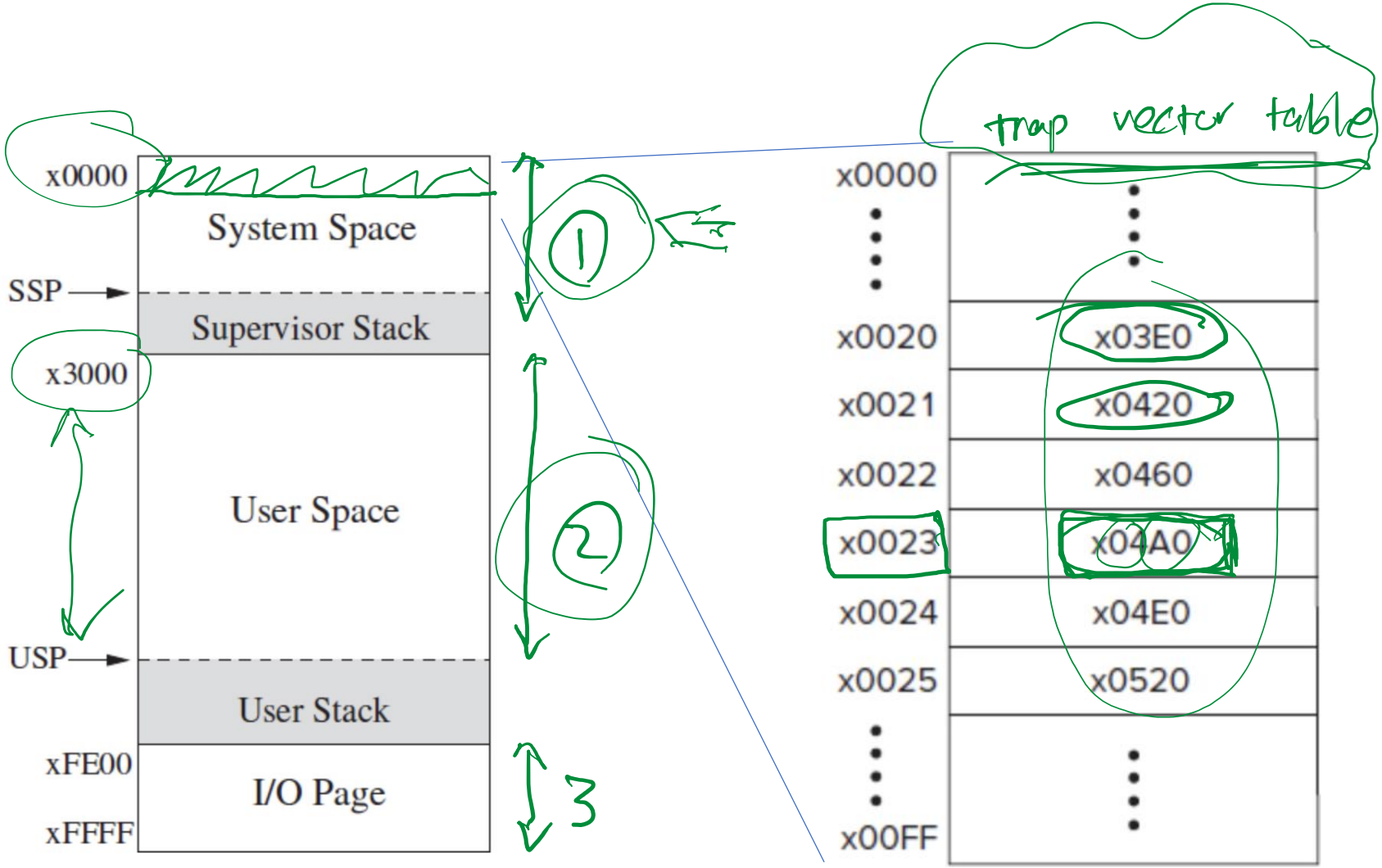
TRAP <trap vector>

- Cause a service routine to execute
  1. Push PSR and PC to the system stack
  2. Set privilege bit in PSR to "0"
     - 0-> supervisor mode ; 1-> user mode
  3. Set PC = mem[ZEXT(trap vector)]
  4. Execute service routine



Example

```
.ORIG x3000
TRAP x23
```



The TRAP Mechanism

- *Service Routine*
- *Trap Vector Table*
- *The TRAP Instruction*
- *A Linkage*

RTI (Return from TRAP/Interrupt) Instruction

- Return control to the calling program

1. Pop two value from system stack
   - PSR and PC

PC ← Restored PC

2. Set PSR bit[15] to 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Pr						PL n							N	Z	P	**PSR**
Priv						Priority							cond codes			

what is value of PSR[15] @ B ?