While:

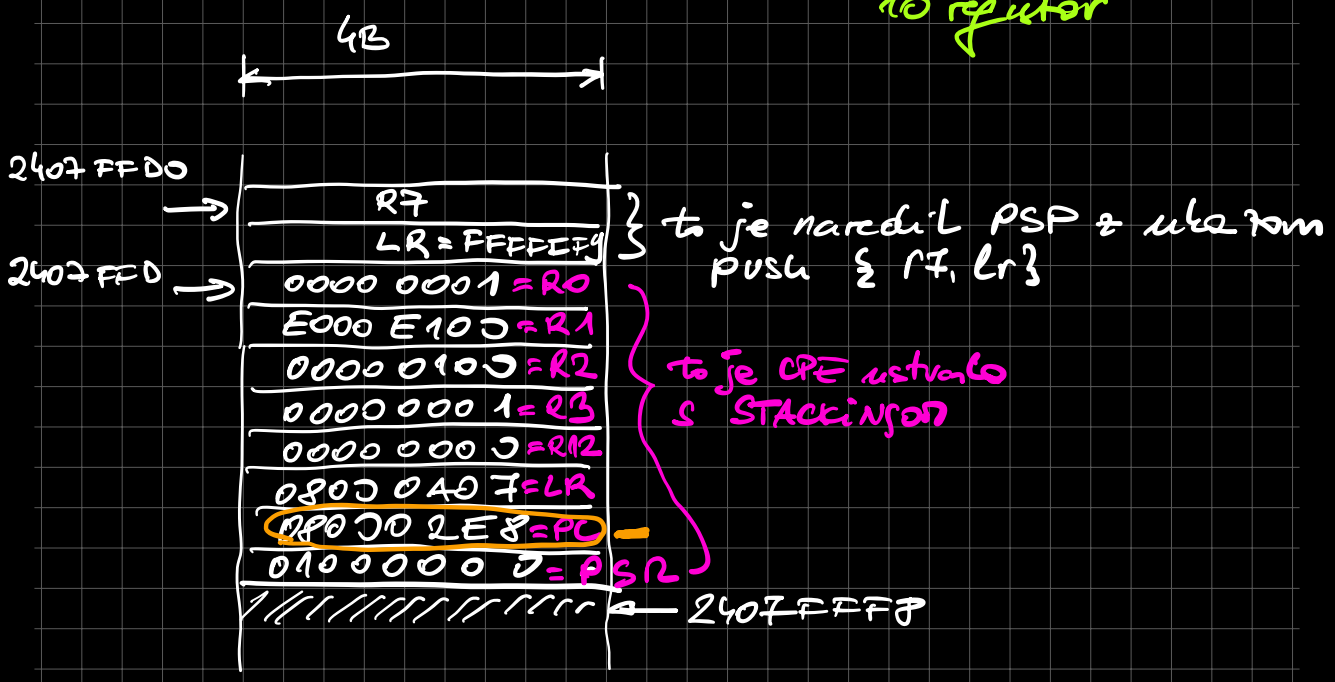
		Po vstopu
R0 =	0x00000001	R0 = 0x00000001
R1 =	0xE000E100	⋮
R2 =	0x00000100	⋮
R3 =	0x00000001	⋮
R4 =	0x24000030	
⋮		
R12 =	0x00000000	
SP =	0x2407FFF8	SP = 0x2407FFD0
LR =	0x08000A07	LR = 0xFFFFFFF9
PC =	0x080002E8	PC = 0x0800055C
xPSR =	0x10000000	PSR = 0x10000038

- stari SP = 0x2407FFF8
- novi SP = 0x2407FFD0

$0x00000028_{16} = 40_{10}$

⇓

Na sklad imamo ostro vsebino 10 registrov



PRVI UKAZ PSP: PUSH R7, LR → FFFFFFF9

Zadnji ukaz PSP:

POP R7, PC → anomalna

PC ← LR

po tem ukazu je SP = 0x2407FFD0

S tem ukazom PSP!! (PC ← FFFFFFF9)

LPE zopet dostopen

S skladu obnovi: R0, R1, R2, R3, R12, LR, PC, PSR

Prekinitve pri procesorjih Intel

- INTR (Interrupt Request)
- NMI (Non-Maskable Interrupt)

Intel ima do 32 pasti in do 224 zunanjih prekinitev:

TIPIČNE PASTI:

ID	
0	Div by 0
1	Debug
2	NMI
⋮	
14	Page Fault
⋮	

vsako prekinitev/past ima svoj ID

⇓

pri Intelu temu rečejo ŠTEVILKA PREKINITVENEGA VEKTORJA (INTERRUPT VECTOR NUMBER)

Ob vstopu v prekinitev:

$$PC \leftarrow M[0x00000000 + 4 \cdot ID]$$

↳ Isto kot Cortex!

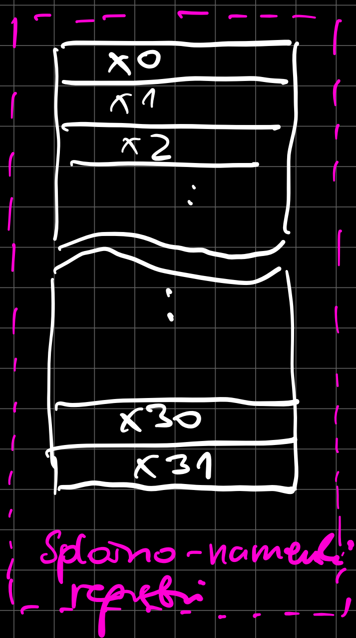
1. Shrani kontekst na sklad
2. Prevzame se vektor:

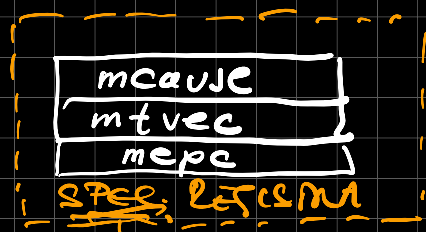
   $$PC \leftarrow M[0x0000\,0000 + 4 * ID)$$

3. začne se izvajat PSP

Prekinitve pri RISC-V

Programski model:





ker se teh registrov ne da naslavljati v splošnih ukazih

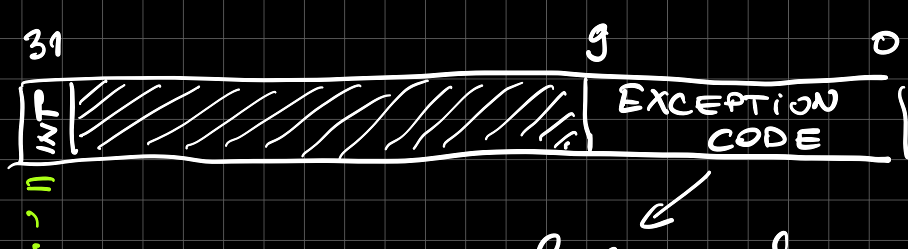
uporabljamo posebne ukaze

CSRR, CSRW,
CSRS, CSRC...

mcause register hrani vzrok prekinitve

mtvec register hrani pravilo, kako tvoriti naslov PSP

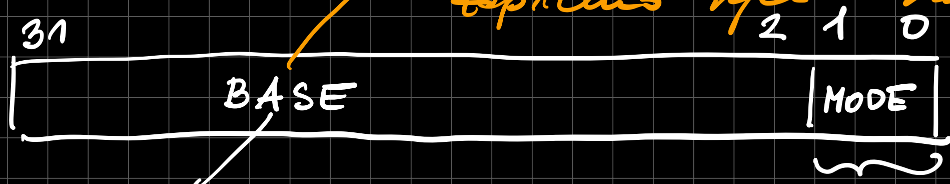
mcause:



koda vzroka prekinitve oz. je to ID prekinitve

mtvec

tabelo vektorjev lahko postavimo kjerkoli v MEM, ker jo zapišemo mi sami



to je začetni naslov prekinitvene tabele

način računanja naslova PSP

Direktni ('00')

PC ← BASE . '00'

zgornjih 30 bitov v PC

⇓

za vse možne prekinitve imamo 1 sam prekinitveno-servisni podprogram

⇓

V tem PSP moramo programsko ugotoviti kdo je prekinil.

⇓

preberemo spodnjih 10 bitov iz mcause registra in se na osnovi njihove vrednosti skoči ..

```
switch (mcause[9:0]) {

    case : 0
        call handler 0;
    case : 1
        call handler 1;
```

Vektorski ('01')

⇓

PC ← BASE + 4x mcause[9:0]

⇓

v PC se zapiše naslov PSP glede na ID prekinitve

Vstop v prekinitev

→ RISC-V ničesar ne shranjuje na sklad!!

⇓

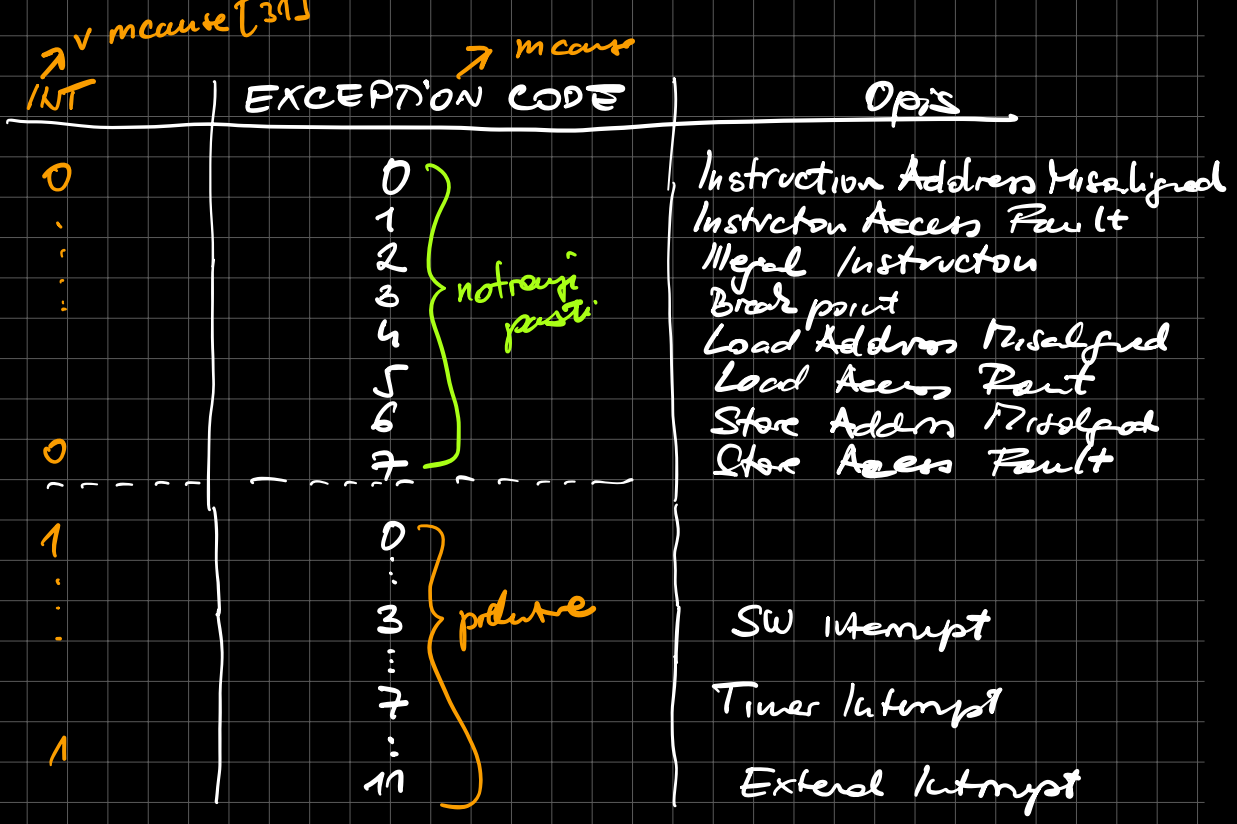
Vse je prepuščeno programerju!!!

1. mepc ← PC // shrani PC
2. PC ← naslov PSP (odvisno od načina ↓ MODE v registru mtvec)
3. začne se izvajati PSP: PSP shrani kontekst na sklad !!!
4. PSP se zaključi z ukazom **mret**

mret: PC ← mepc

5. Nadaljuje se izvajanje prej prekinjenega programa

Prekinitve in past. pri RISC-V:



Prekinitvena tabela

```
BASE :  j   handler 0   ; 0
        j   handler 1   ; 1
        j   handler 2   ; 2
        j   handler 3   ; 3
        .
        .
        .
        j   handler 7   ; 7
        .
        .
        .
        j   handler 11
        .
        .
        .
```

Handler 7:

beri mcause;

če je mcause[31] = 0:

; TrapHandler

sicer ; IntHandler

TrapHandler:

IntHandler: