

VGRS, 3.10.2023

pa3cio@fri.uni-lj.si

rok.cesnovar@fri.uni-lj.si

HAIT : projektna naloga

~~3 aprila 102 v STUDIO~~

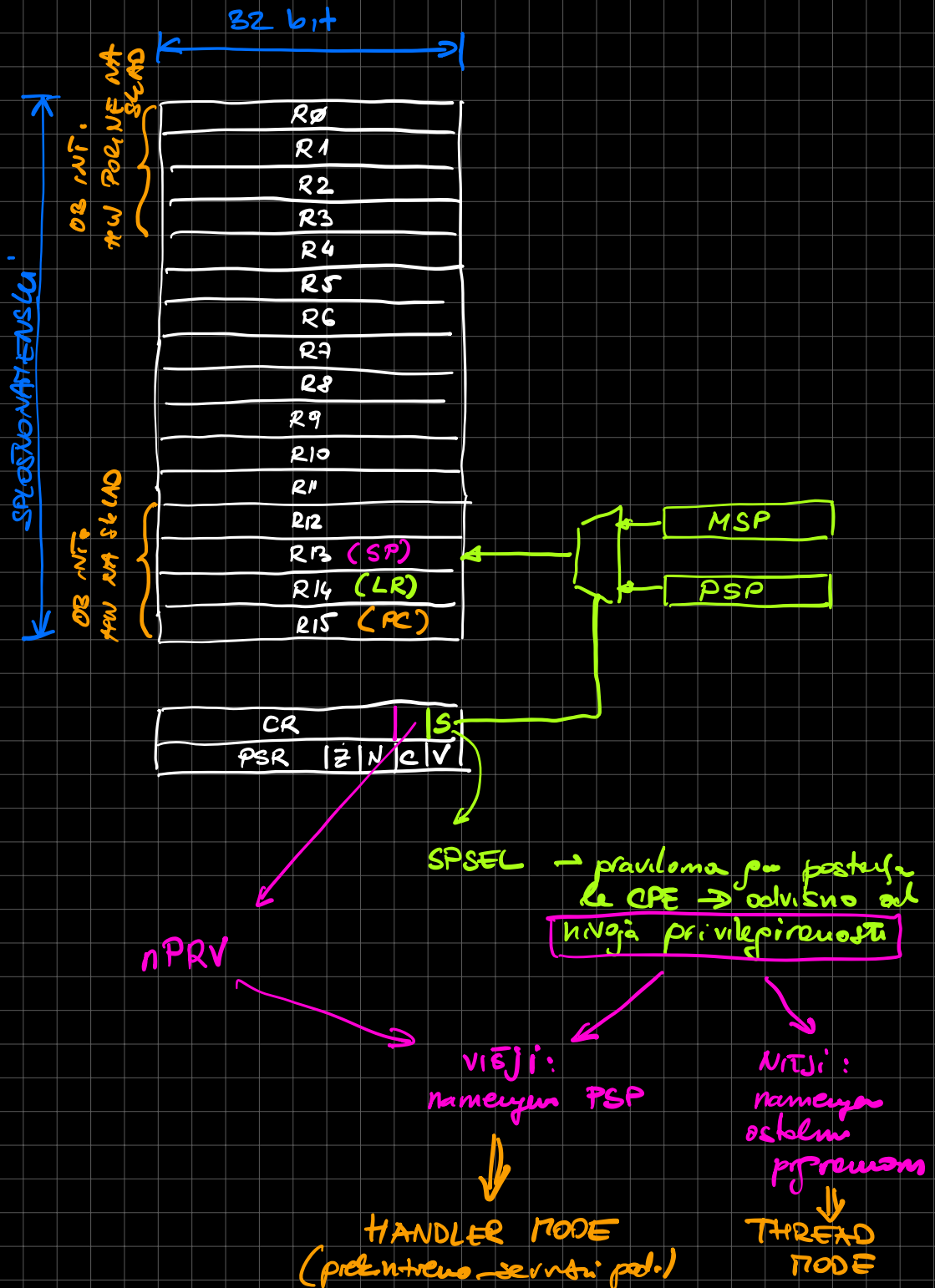
Sklepi literature

- predavanja in vaj
- moji zapiski
- gradivo za vaj
- moji osnutki učbenka
- učbenki tujih avtorjev

STM 32 H7 : Cortex M7 cip

Prelimīnares pri: ARM-Cortex

Programistu modelis CPE:



SKLAD

ARM ABI :

FULL DESCENDING

SP kaže na
zadnji vskujen
podatek

SKLAD NARAJETA
V SP-ovi POMOZNI
NASLOVI

PUSH reg :

$SP \leftarrow SP - 4$ (ADDI R13, R13, -4)

$M[SP] \leftarrow \text{reg}$ (SW O(R13), reg)

POP reg :

$\text{reg} \leftarrow M[SP]$ (LW reg, O(R13))

$SP \leftarrow SP + 4$ (ADDI R13, R13, 4)

UKAZI ZA DELO S SKLADOM

LDMFD (\approx pop)

STMFD (\approx push)

Vstop v prekinitev

→ ko procesor dobi prekinitevno zahtevo ⇒
CPE vs, kedo je prekinil ⇒ (ID vira)
↓
1...254

1. Poveže učne ad vključno stopnji EX v cevovodu

2. CPE hardware (brez kvazija olasa!!!)
na sled porine naslednje registre:

Hardware
Stacking

PSR, PC, LR, R12, R3, R2, R1, R0

→
v tem vrstnem redu se porinjo na sled

del konteksta, & se few shrepi in prapoumeja & to ni treba shrepi.

CPE sama opre:

3. LR ← 0xFFFFFFFF_x } hej, v prekiniti sem 😊

4. PC ← M [4 * ID]

→ označi nivo privilegijosti in sled prekinitevne procesne

prekinitevno servo: podprogram

Urniker it prekinitveņa servitveņa palproģams:

`pc ← Lr;`

CPE mēde fa elvā mē
slādeē mē:

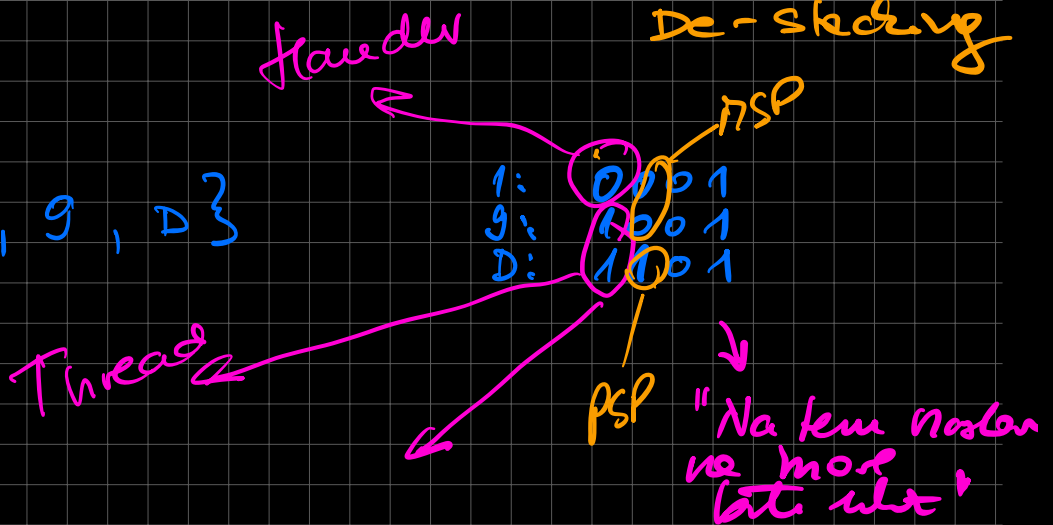
Ē jē v Lr = 0xFFFFFFFFx

Sec:
`pc ← Lr`

```
POP r0
POP r1
POP r2
POP r3
POP r12
POP Lr
POP pc
POP psp
```

Hardware
De-stacking

$x \in \{1, g, D\}$



1: Handler Node + MSP

g: Thread Node + MSP

D: Thread Node + PSP

Tabela prekinjenih vektorov: (veštarsko tabelo)

začne se na naslovu 0x0000 0004 ..

0x0000 0000: začetna vredost SP

0x0000 0004:	naslov	PSP _e	z	ID1
0x0000 0008:	naslov	PSP _i	z	ID2
0x0000 000C:	naslov	PSP _j	z	ID3

⋮

↓
vešarska tabela