

PRED INT:

R0 = 0x0000 0001
 R1 = 0xE000 E100
 R2 = 0x0000 0100
 R3 = 0x0000 0001
 :
 R12 = 0x0000 0000
 LR = 0x0800 9BB
 PC = 0x0800 2E8
 PSR = 0x01000000

 SP = 0x2407 FFF8

PO INT:

SP = 0x2407 FFd0
0x28 = (40₁₀)

sklad se je povečal za 40 B, kar je 10 register

Slika sklada po vstopu v PSP:



↓
tako se je moglo manj

PREKLAPANJE OPRAVIL

upotabili smo:

interrupt entry:

1. USTVARITI SE **HW Stack Frame**
2. $LR \leftarrow 0xFFFFFFFF$

HW stack je kontekst prekinjenega programa

interrupt exit:

1. $PC \leftarrow LR$ ($0xFFFFFFFF$)
↓
POSLEDICA JE:
HW destacking

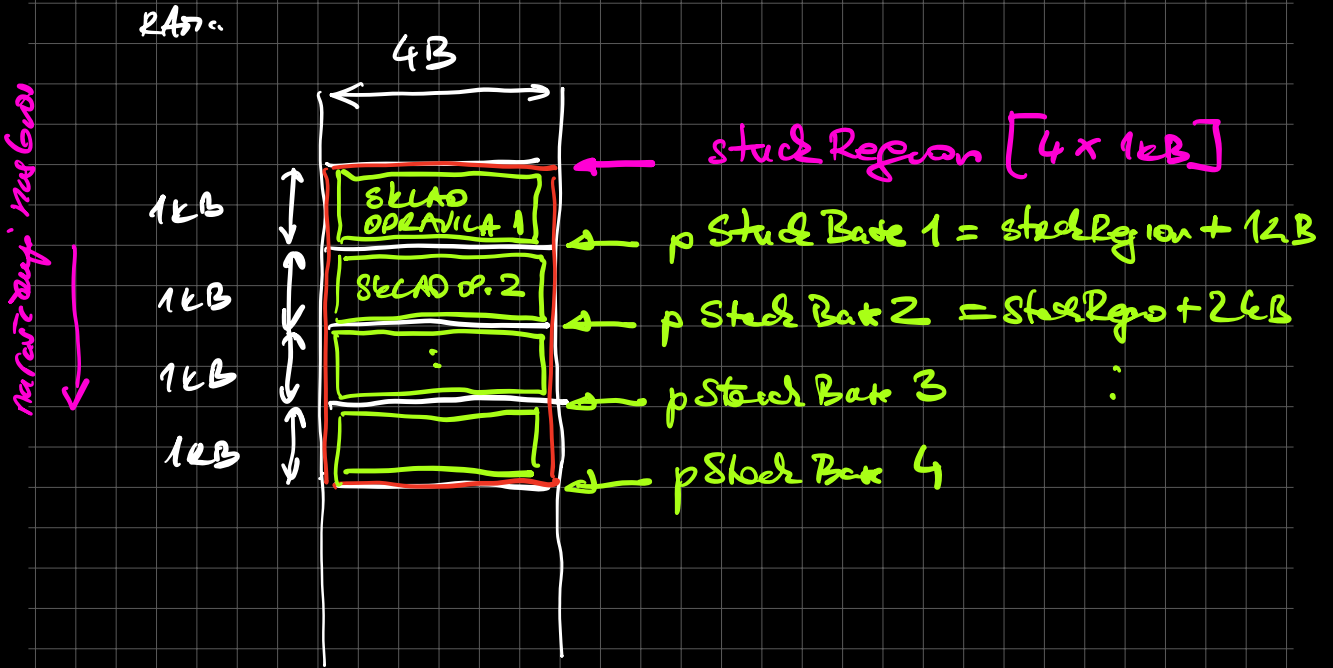
IDEJA PREKLAPANJA OPRAVIL

↓
PSP preprosto zamenjaj s sledovno
↓

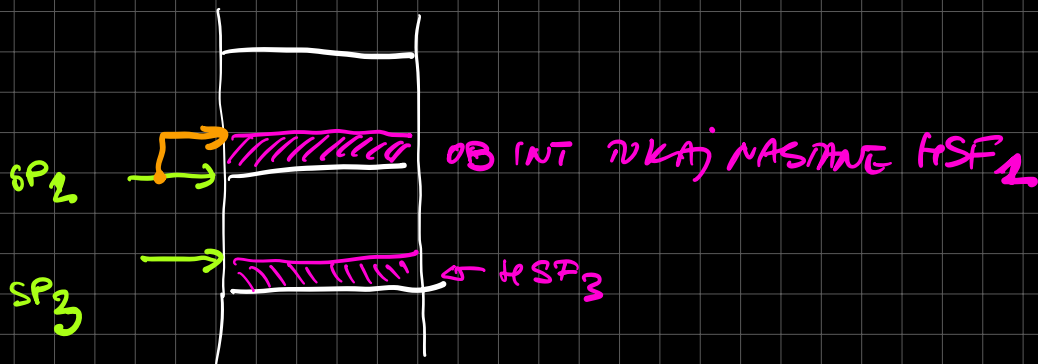
ta naj tedaj kaže na HSR
procedure (opravilo) v
katero se želimo vrniti

TEHNI (predpostavimo 4 opravila)

→ vsako opravilo bo imelo svoj sklad!



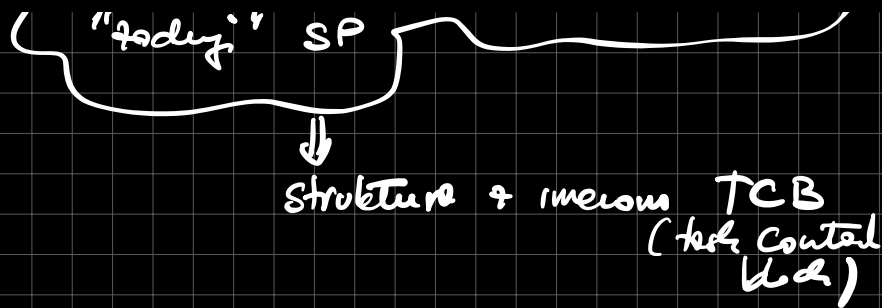
Priloga k opravilu 2:



TASK SWITCH:

SP₁ → SP₃ → to mora nast. PSP

Za vsako opravilo moram v pomnilniku imeti znanen kraj hrota. VSAJ njega!



→ HSE vseh opravil moramo na 70% delu
ustvariti sami:

r0: ← 0
r1: ← 0
r2: ← 0
r3: ← 0
r12: ← 0
lr: ← 0
pc: ← naslov 1. elne opravila
psr: ← 0

→ noben tega opravila ne pollece !!!

opravila se
ne del: ne
one zafra!

while (1) !!

