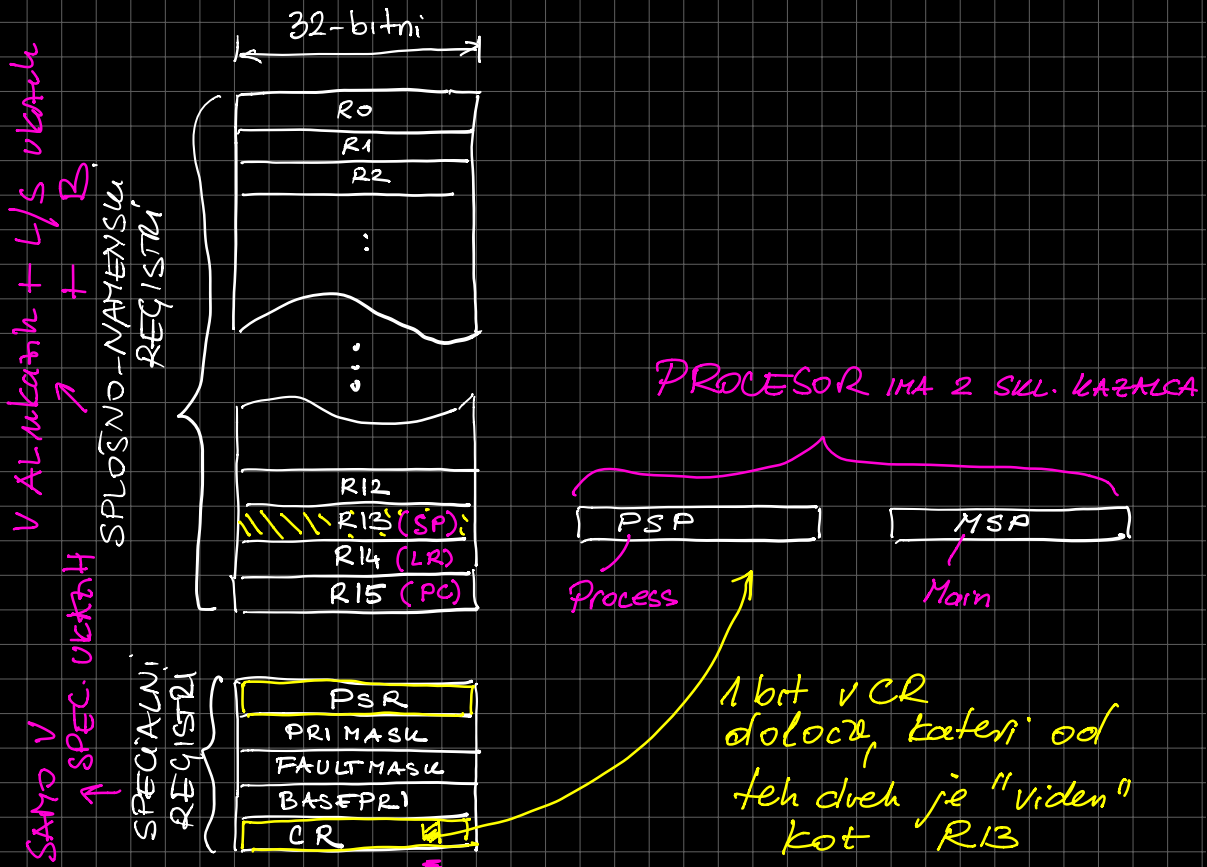
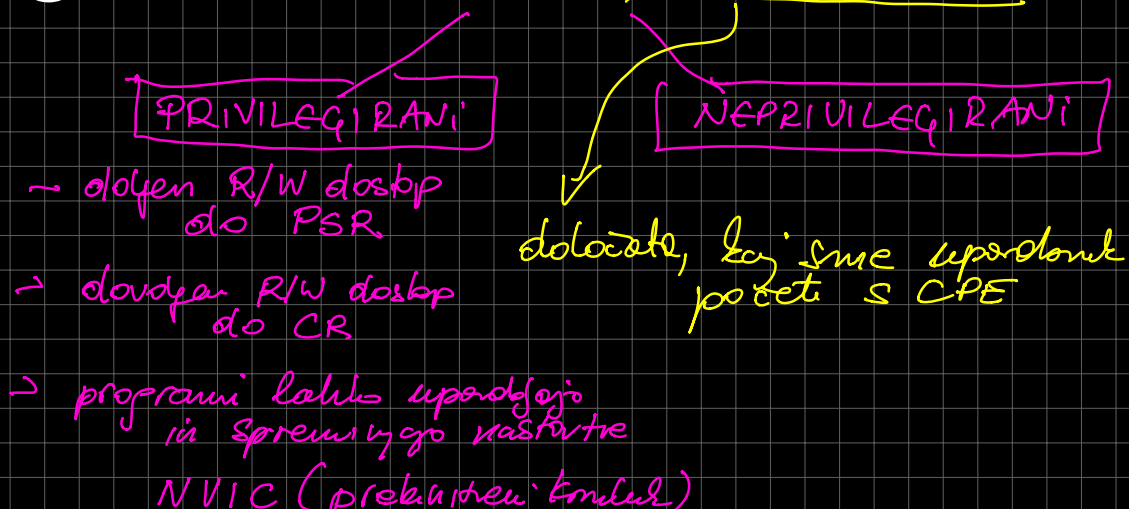


ARM Cortex-M

① PROGRAMSKI MODEL



② Cortex-M IMA DVA NAČINA DELOVANJA



→ dostopajo do vseh pam. bled
v pomnilniku in do vseh
registrarskih vseh v/i naprav

↓
sistemске procedure

↓
uporabne procedure

③ CORTEX-M IMA ŠE 2 NAČINA DELOVANJA

Handler Mode

↓
Prekinitevno-servisni
podprogrami

→ v ta način CPE vstopi
ob prekinitvah in
izjemah

↓
VEDNO PRIVILEGIRANI!

Thread Mode

↓
Vsi ostali podprogrami

→ CPE vslopi po Resetu

↓
Lahko je PRIVILEGI ali
NEPRIVILEGI

↓
ddoča en bit v CR

POHETIBNO!

→ edini način, da se pregreva, k' se
izbrzo v THREAD načinu in v
NEPRIVILEGIRANEM načinu. Prestopimo v
PRIVILEGIRANI NAČIN je, da pridemo
prekinitve ali izjeme in v prek.
servisnem podprogramu "popenimo" bit v
CR, ki ddoča privilegiranost

✓
PREKINITVE/IZJEME bodo edini dovoljeni načini
za uporabo sistemskih sredstev

③ SKLAD V CORTEX-M

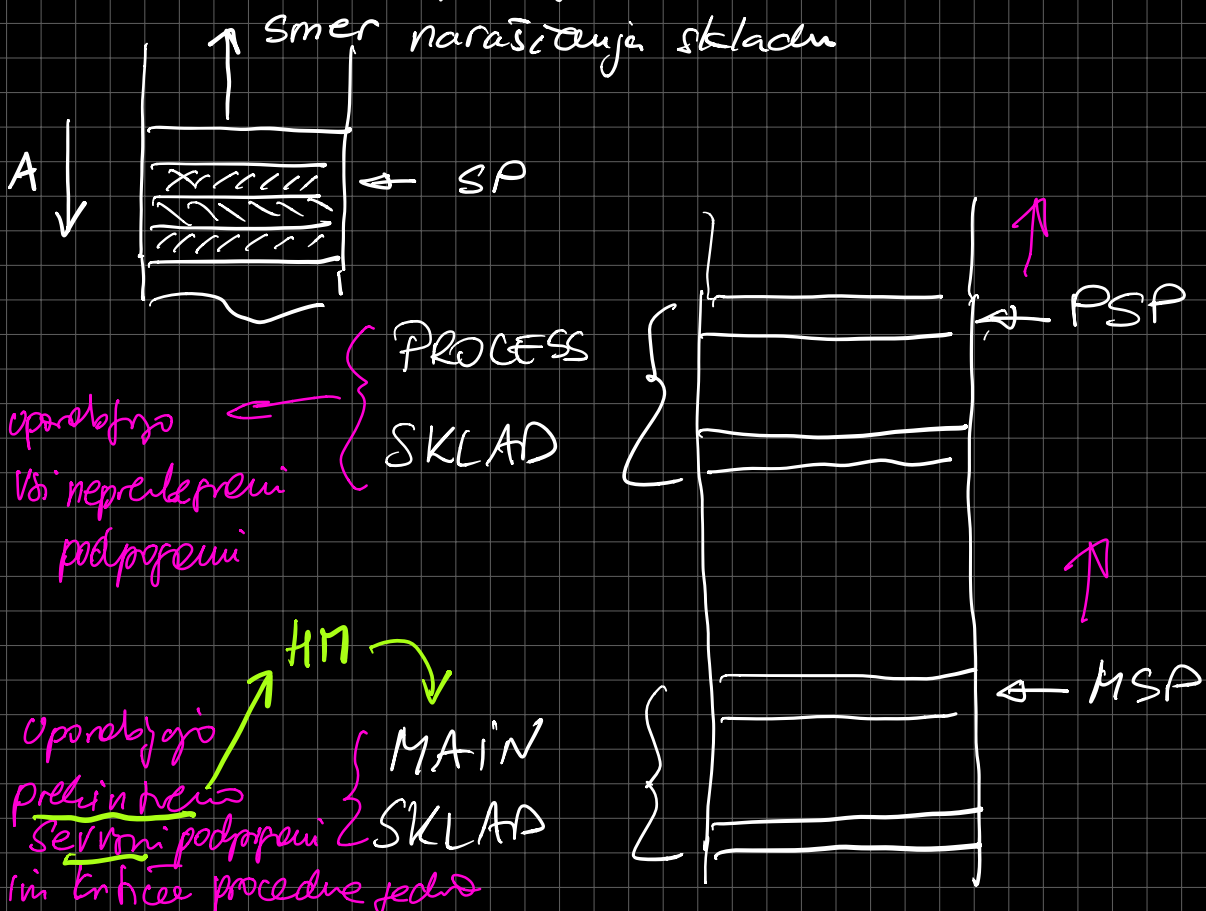
→ Cortex-M uporablja dva skladovna koralca: PSP in MSP

→ lahko izolirujemo dva ločena sklada v pomnilniku

→ na en sklad kaže PSP, na drugi sklad kaže MSP

→ SKLAD: Full Descending (FD)

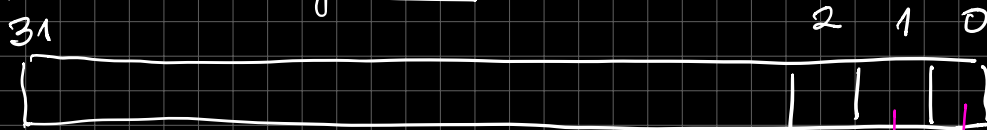
1. SP kaže na zadnji podatek na skladu
2. sklad narašča v smeri padajočih naslovov



CPU Mode	Uporaba	Nivo privilegiranosti	SKLAD
THREAD	Uporabniške procedure	→ odvisno od CR (NEPRIV)	→ odvisno od CR (PROCESS)
HANDLER	Prekinitveno - Servisni podprogrami	PRIVILEGIRANI	MAIN

ololo omoedino it PSP

Kontrolni register:



MSP : 0 SPSEL

PSP : 1

PRIVIL : 0 nPRIV

NEPRIV. 1

PREKINITIVE PRI CORTEX-M

IZJEME = PREKINITIVE & PASTI
(EXCEPTIONS)

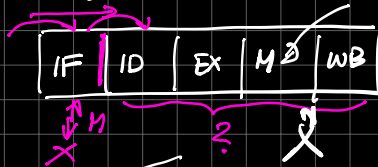
(*) MEHANIZEM OBRAVNAVE IZJEM V CPE (Odziv CPE na izjeme)

Ko se zgodí izjema in se CPE odzove:

① CPE ustavi izvajanje trenutno izvajajočega se programa
kako?

a) prenehajoče delo iz pomnilnika

b) ker je cevovod:



Wi' ukoni v cevovodu
1redgo do konca
↳ traji (4 up.)

LATENCY

Ne dokončen
novejšo delo
v cevovodu

→ TETIVA
- PC moramo
potem popravit
zo nuj.

→ EX sprejeti ??
zadržica ??

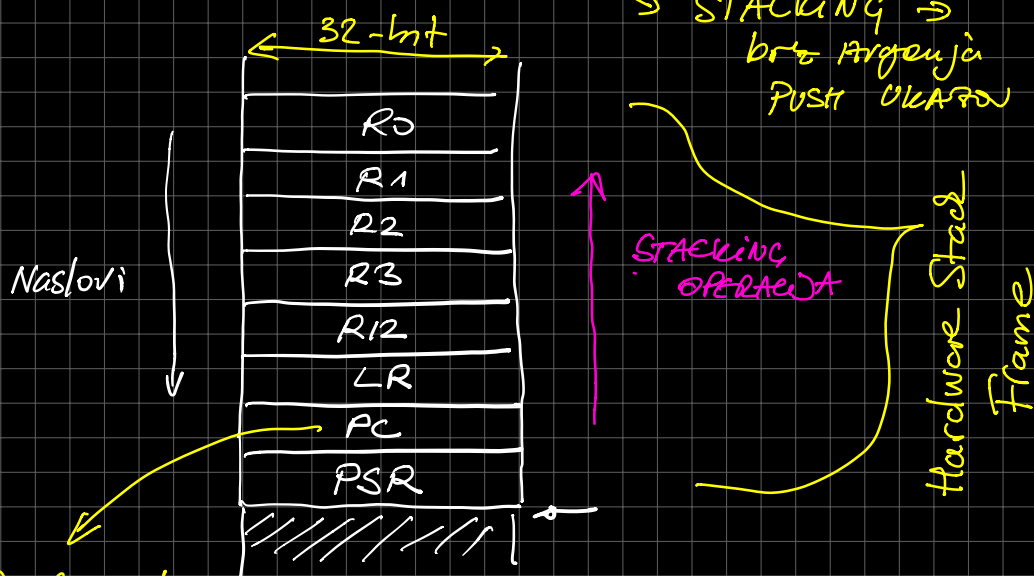
→ =

hvedemo do konca
delo, ki so v EX, M, WB
Latency = 3

② CPE shranjuje stanje prekinjenega programa

→ SKLAD!!

→ Cortex začne shranjevati naslednje registre na klad



POVRATNI

NASLOV V PREKINJENEM PROGRAMU

③ CPE začne z izvoženjem PSP-a:

PC ← naslov PSP ? → NASLEDNJIČ

LR ← FFFFFFFFxx

sedaj se začne izvajati delo PSP-a

④ VRNITEV V PREKINJENI PROGRAM

→ Treba je obnoviti vse registre, u to
na HW stack frame-u

Treba jih je prebrati (prebrati) s sklada

PSP z
ukm POP

CPE ✓

Kako naj CPE ve,
kdy naj začne z
opracjo DESTACKING
???

NA koncu PSP-a mora biti nek ukm, ki
ji to nakazuje (IRET)

Običajno pismo v C-ju

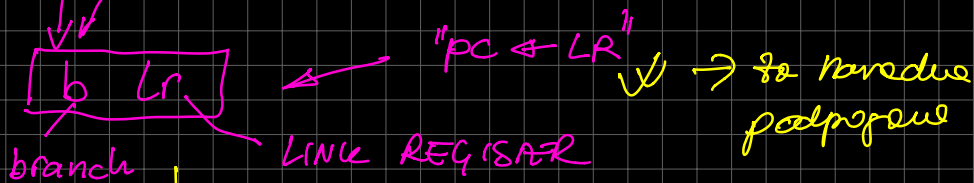
POSEBNA REZERVIрана BESEDA
void MgaPreznanjePolno () {
--int
for...

Kako NAJ PREVAJALNIK POVEŠ, DA TRAJAJ
DA IRET UKAZ
??

ARM Cortex imajo posebno rešitev

zadnji ukaz v PSP je ena

zadnjemu ukazu v navadnem
podprogramu (ni IRET ukaza)



ANPAK: PSP mora sprožiti DESTACKING!!
in ne PC ← LR

ARM Cortex:

Če je v PC $\leftarrow$ $\overbrace{\text{FFFFFF} \times \times}^{24 \text{ ena}}$

CPE ne more ukaz

Stežer nastava

↓

ZACNE DESTACKING
OPERACIJO