



Kazalec argumentov AP se uporablja za prenos argumentov pri klicu podprograma in ga na enak način uporabljajo vsi višji programski jeziki. Pri klicu podprograma pa procesor sam poskrbi, da AP res kaže na pravi naslov.

Tudi FP se uporablja za delo s podprogrami. Ta register kaže kje je podatkovna struktura, ki se imenuje "call frame" in v kateri so zapisane vrednosti registrov pred klicem in informacija operacijskemu sistemu kaj naj naredi, če pride do napake pri izvajanju podprograma.

Register SP služi za organiziranje sklada, to je dinamična podatkovna struktura. Količina pomnilnika, ki ga sklad zasede je vedno ravno enaka skupni velikosti vseh podatkov v skladu. Sklad je podoben skladovalni drv ali kupu krožnikov. Če hočemo shraniti krožnik, ga položimo na vrh kupa. Zadnji dodani krožnik je na vrhu, prej dodani po se po vrsti pod njim. Ko vzamemo krožnik s kupa, vzamemo najprej tistega, ki smo ga nazadnje postavili na kup, nato vzamemo predzadnjega in tako naprej. Taka struktura nosi angleško kratico LIFO od "last in first out", kar pomeni zadnji noter, prvi ven. Kazalec sklada kaže vedno na podatek, ki smo ga nazadnje zapisali v sklad in ta podatek dobimo, ko vzamemo nekaj s sklada. Ukazi za delo s skladom sami poskrbijo, da se kazalec sklada sproti prilagaja vsem spremembam v skladu.

Za razliko od ostalih treh registrov, ki se spremenijo le, ko delamo s skladom ali pri klicu podprograma, se programski števec spremeni po vsakem ukazu. Ta register kaže vedno na naslednji ukaz in procesor sam popravi to vrednost po vsakem prebranem ukazu. Tesa registra torej ne moremo uporabljati za shranjevanje svojih podatkov.

Tudi pri ostalih dvanajstih registrih se nam lahko zdi, da vrednost v registru ni takšna, kot jo pričakujemo. Po show time show time dogovoru uporabljajo višji programski jeziki in sistemski podprogrami registra R0 in R1 za vračanje statusa, s katerim se je končalo izvajanje podprograma ali za rezultat, če kličemo podprogram kot funkcijo. Po klicu sistemskega podprograma bomo torej imeli v registru R0 drugačno vrednost kot pred klicem.

Drugi primer, ki nam lahko pokvari stanje v registrih je uporaba ukazov za delo s teksti. Ti ukazi namreč uporabljajo registre R0 do R5 za shranjevanje števecov znakov v tekstih in za kazalce na tekoče znake. Pred uporabo teh ukazov moramo spraviti vsebino teh šestih registrov na varno, npr. na sklad, po uporabi pa staro vsebino vrnemo v registre.

1.1.2 Stanje procesorja

Trenutno stanje procesorja je zapisano v posebnem registru z imenom "processor status longword" ali krajše PSL. Tudi ta register ima 32 bitov, ki pa so razdeljeni na dve skupini po 16. Biti od 0 do 15 vsebujejo nepriviligirane podatke, ki so dostopni vsakemu