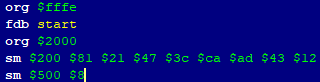
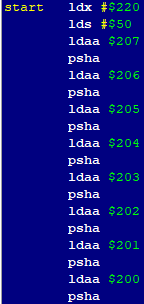
## ­Potek dela

Vsak program se začne z standardnimi ukazi – org ter fdb. Za tem, sem rabil na naslov $200 shraniti svojih 8 števil. Izbral sem števila $81, $21, $47, $3C, $CA, $AD, $43, $12, pri čemer pomeni simbol $, da je število v šestnajstiškem sistemu. Na naslov $500 sem shranil številko 8, ki je kasneje pripomogla pri uporabi zanke.



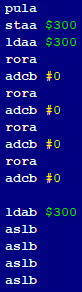
Slika : Začetek programa

Nato se program začne. Nastavil sem indeksni register na naslov $220. Ta je bil kasneje uporabljen za shranjevanje števil. Označil sem dno sklada na naslov $50. Nato sem iz naslova $207 naložil svojo številko v akumulator A. To število sem nato shranil v sklad. Enako sem nato naredil za število 206-200. To sem naredil, da sem lahko to števila kasneje pridobival ven iz sklada. V napačnem vrstnem redu sem jih zapisoval, saj sklad deluje na principu »last in first out« in bi se vrstni red števil obrnil, če bo jih zapisal od 200-207, namesto od 207-200.



Slika : Zapisovanje v sklad

Nato pridobim iz sklada prvo število. Shranim ga na naslov $300, da ga lahko kasneje uporabim. Da se prepričam, da bo vse delovalo pravilno, ga spet naložim iz naslova $300, čeprav ta postopek ni potreben. To število nato štirikrat rotiram v desno, pri čemer po vsaki rotaciji pobrišem carry bit, da se ne bi prištevala dodatna števila nazaj v moje število. S tem sem prve štiri bite števila premaknil na desno stran. Nato v akumulator B naložim isto število, tudi iz naslova $300. Tega štirikrat shiftam v levo na podoben način, kot sem prej rotiral prve štiri bite. S tem se zadnje štirje biti števila premaknejo na levo stran števila.

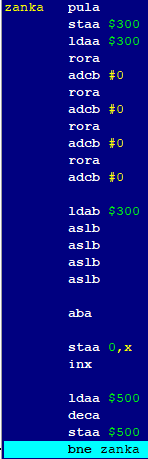


Slika : Premikanje bitov

Sedaj imamo v akumulatorju A prve štiri bite števila na desni strani, ter v akumulatorju B zadnje štiri bite na levi strani. Akumulatorja skupaj seštejemo in dobimo končno število. To nato s pomočjo indeksnega registra shranimo na prej označeno lokacijo, $220. Indeksni register povečamo za 1.

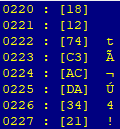
Nato naložimo v akumulator A vrednost naslova $500, jo zmanjšamo za 1, in se ta shrani na enak naslov. To število nato primerjamo s številom 0. Če vrednost števila ni 0, se zanka ponovi. Ponovila se bo 8-krat, saj smo na toliko nastavili vrednost naslova $500.

Ker uporabljamo sklad, se pri naslednji ponovitvi naloži naslednjo število in se proces ponovi.



Slika : Zanka

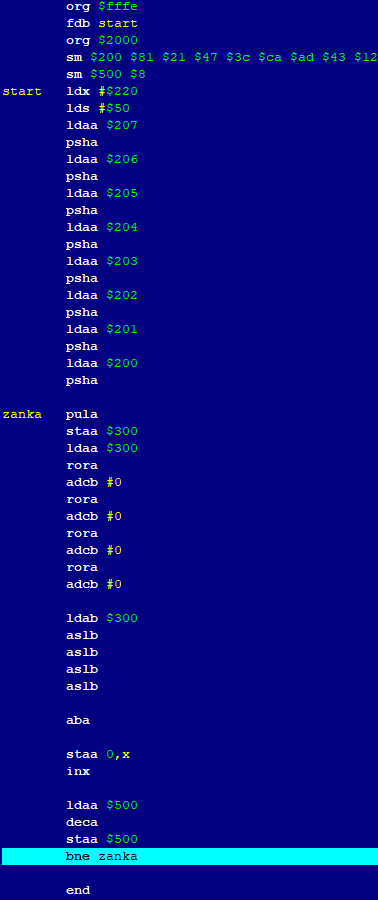
S tem je program končan. Na naslovih $220 in dalje so sedaj zapisana naša obrnjena števila. Tukaj so rešitve:



Slika : Rešitve

S tem je dokazano, da program deluje pravilno, saj štirje biti ponazarjajo eno število v šestnajstiškem sistemu. Torej, se ptevilo 81, obrne v 18.

Tu je še slika celega programa:



Slika : Celoten program