

Kodi nekaterih operacij za ukaz RST 28

<u>01</u>	<u>zamenjaj</u>	Zamenjaj zgornji dve števili na skladu.
<u>02</u>	<u>odstrani</u>	Odstrani zgornje število s sklada.
<u>03</u>	<u>odštej</u>	Odstrani zgornji dve števili in porini na sklad ... njuno razliko (odšteje zadnje število od predzadnjega).
<u>04</u>	<u>pomoži</u>	... njun zmnožek.
<u>05</u>	<u>deji</u>	... rezultat deljenja predzadnjega števila z zadnjim.
<u>06</u>	<u>potenciraj</u>	... predzadnje število na potenco zadnjega.
<u>0F</u>	<u>seštej</u>	... njun seštevek.
<u>17</u>	<u>z-seštej</u>	podobno kot "seštej", le za seštevanje znakov.
<u>18</u>	<u>VAL\$</u>	Nadomesti zgornje število z njegovim VAL\$.
<u>19</u>	<u>usr_z</u>	Nadomesti zgornjo vrednost na skladu (zaporedje znakov) z vrednostjo USR tega zaporedja.
<u>1B</u>	<u>neg</u>	Nadomesti zgornje število z njegovo negativno vrednostjo.
<u>31</u>	<u>podvoji</u>	Porini na sklad še enkrat, isto vrednost, ki je trenutno na vrhu.
<u>3B</u>	<u>končaj</u>	Vrni se v normalni strojni program.
<u>A0</u>	<u>konst 0</u>	Porini na sklad število 0.
<u>A1</u>	<u>konst 1</u>	Porini na sklad število 1.
<u>A2</u>	<u>konst 1/2</u>	Porini na sklad število 1/2.
<u>A3</u>	<u>konst PI/2</u>	Porini na sklad število PI/2 (1.5707963).
<u>A4</u>	<u>konst 10</u>	Porini na sklad število 10.
<u>Cn</u>	<u>spravi_n</u>	Spravi vrednost z vrha sklada v prostor n računskega pomnilnika.
<u>En</u>	<u>vzemi_n</u>	Porini na sklad vrednost v prostoru n računskega pomnilnika.

Kodi, ki sledijo, nadomestijo vrhno vrednost (oz. zaporedje znakov) na računskem skladu z rezultatom ustrezne operacije. Pomen operacij je enak kot v osnovi.

<u>1C</u>	<u>code</u>	<u>20</u>	<u>cos</u>	<u>24</u>	<u>atn</u>	<u>28</u>	<u>sgf</u>	<u>2C</u>	<u>in</u>
<u>1D</u>	<u>val</u>	<u>21</u>	<u>tan</u>	<u>25</u>	<u>ln</u>	<u>29</u>	<u>sgn</u>	<u>2D</u>	<u>usr_št</u>
<u>1E</u>	<u>len</u>	<u>22</u>	<u>asn</u>	<u>26</u>	<u>exp</u>	<u>2A</u>	<u>abs</u>	<u>2E</u>	<u>str\$</u>
<u>1F</u>	<u>sin</u>	<u>23</u>	<u>acs</u>	<u>27</u>	<u>int</u>	<u>2B</u>	<u>peek</u>	<u>2F</u>	<u>chr\$</u>

ime podprograma      naslov      učinek

<u>STACK_A</u>	<u>2D28</u>	pretvori celo število v registru A v 5-zložno obliko in ga porine na sklad.
<u>STACK_BC</u>	<u>2D2B</u>	enako kot <u>STACK_A</u> , vendar za število v registrskem paru BC.
<u>STACK_AEDCB</u>	<u>2AB6</u>	porine na sklad vrednost registrskega peterčka AEDCB.

Podprogrami za pobiranje s sklada so podobni:

ime podprograma	naslov	učinek
<u>FP_TO_A</u>	<u>2DD5</u>	naloži število z vrha računskega sklada v register A.
<u>FP_TO_BC</u>	<u>2DA2</u>	naloži število z vrha računskega sklada v registrski par BC.
<u>FP_TO_AEDCB</u>	<u>2BF1</u>	naloži število z vrha računskega sklada v registrski peterček.

Ob klicu FP\_TO\_AEDCB ne bo nikoli težav - 5 zlogov lahko vedno zložimo v 5 registrov. Drugače je z FP\_TO\_A in z FP\_TO\_BC. Na vrhu sklada je lahko število, ki je preveliko za v en register ali za v registrski par. To ugotavljamo z zastavicami. Če pri prelaganju ni nobenih težav, bo ničelna zastavica dvignjena, zastavica prenosa pa bo spuščena. Če je zastavica prenosa dvignjena, je bila absolutna vrednost števila prevelika. Če je ničelna zastavica spuščena, je bilo število negativno. V tem primeru bomo v registru kljub temu dobili pravilno absolutno vrednost števila.

V RAM-u Spectruma najdemo 30 zlogov prostora, ki je opisan v priročniku med sistemskimi spremenljivkami in nosi ime MEMBOT. To je računski pomnilnik. V njem je prostora za 6 vrednosti v plavajoči vejici. Z ukazom RST 28 lahko uporabljamo dva niza podatkovnih kod: Cn ter En, pri čemer ima n vrednost od 0 do 5. Niz C spravi vrednost z vrha računskega sklada v prostor n (vrednosti na skladu pri tem NE odstrani). Niz E vzame vrednost prostora n in jo porine na sklad.

Pri uporabi računskega pomnilnika je treba upoštevati dve opozorili. Nekateri funkcije (npr. SIN, COS, STR\$, če omenimo samo tri) izbiršejo vrednosti v prostorih 0, 1 in 2 računskega