

tipke. Pomembno je, da se v tem primeru "caps shift" in "symbol shift" NE razlikujeta od ostalih tipk. Ko boste pritisnili samo "symbol shift", bo rezultat FF18 - ravno tako kot bi bil rezultat FF1E, če bi pritisnili samo "S".

Zdaj pa vzemimo, da sta naenkrat pritisnjeni "caps shift" in še neka druga tipka. V tem primeru vrednost v D ne bo FF temveč 27 - kar je kod tipke "caps shift". V E bo kod druge pritisnjene tipke. Podobno velja, če je istočasno pritisnjena "symbol shift" in katerakoli druga tipka - razen "caps shift". V tem primeru bo D vseboval 18 (kar je kod tipke "symbol shift"), medtem ko bo v E kod druge tipke. Če pa istočasno pritisnete "caps shift" in "symbol shift", boste v DE dobili 2718. Zapomniti si velja, da je "caps shift" pred "symbol shift"; če namreč pritisnete obe naenkrat, dobite vrednost 2718 in ne 1827.

KEY SCAN vrne še en pomemben podatek - ničelno zastavico. V vseh omenjenih primerih bo ničelna zastavica ob povratku iz podprograma dvignjena. Če pa se zgodi, da je naenkrat pritisnjenih preveč tipk, bo ničelna zastavica ob povratku spuščena. Spuščena ničelna zastavica torej sporoča, da je vrednost v DE brez pomena. Zato je pred uporabo vrednosti iz registrskega para DE vedno dobro pogledati ničelno zastavico.

Podprogram KEY\_SCAN ima eno pomankljivost - spremeni vrednosti registrov A, B, C, D, E, H in L! Če želimo ohraniti vrednosti parov BC in HL, si pomagamo s skladom:

```
C5    PUSH BC
E5    PUSH HL
CD8E02 CALL KEY_SCAN
E1    POP HL
C1    POP BC
```

Zdaj pa naprej. Vzemimo, da ni bilo pritisnjenih preveč tipk. V naslednjem koraku moramo ugotoviti, ali je bila uporabljena kakšna "prava" tipka. Zavreči moramo torej rezultate FFFF (nobene uporabljene tipke), FF27 (pritisnjena samo "caps shift") in FF18 (pritisnjena le "symbol shift"). V ta namen uporabimo podprogram KEY\_SCAN, na naslovu 031E. Če je v DE "smiselna" vrednost, bo KEY\_TEST najprej napravil dvojice - vrednost registra D bo prenesel v register B, - v register D bo naložil 0.

Nadaljeval bo v eno od dveh smeri:

- če je v DE FFFF, FE27 ali FF18, bo v register A naložil FF, 27 ali 18 in spustil zastavico prenosa;
- če je v DE kakršnakoli druga vrednost, bo v register A naložil kod osnovnega znaka tipke in dvignil zastavico prenosa.

Osnovni znaki tipk so označeni na skici tipkovnice. Njihove kode (t.im. ASCII kode) dobite v basicu z ukazom CODE. Primer: osnovni znak prve leve zgornje tipke je "I", kod tega znaka pa je 31H. Podprogram dobi te vrednosti v posebni tabeli (KEY\_TABLE na naslovu 0205).

Oglejmo si nazadnje še tretji podprogram, ki se imenuje CODE (naslov 0333). Ta ugotovi, kateri znak (ne tipka!) je bil uporabljen in naloži njegov ASCII kod v register A. V ta namen morajo biti izpolnjeni določeni pogoji:

1) Register E mora vsebovati kod osnovnega znaka.

2) Register B mora vsebovati

- FF, če ni pritisnjena nobena tipka "shift",
- 27, če je pritisnjena "caps shift" ter
- 18, če je pritisnjena "symbol shift".

3) Za G-mode (grafični znaki) mora register C vsebovati 02;

za E-mode (znaki nad tipkami) mora C vsebovati 01;

za F-mode (znaki pod tipkami) mora C vsebovati 00 in D 00;

za L-mode (običajni način), mora C vsebovati 00, D 08,

bit 3 sistemske spremenljivke FLAGS\_2 pa mora biti 0;

za C-mode (tiskane črke) mora C vsebovati 00, D 08, bit 3

sistemske spremenljivke FLAGS\_2 pa mora biti 1.

Trenutno uporabljeni znak ugotovite tako, da v register C naložite vrednost sistemske spremenljivke MODE, v D pa sistemske spremenljivke FLAGS ter uporabite KEY\_CODE.

S pravkar pridobljenim znanjem lahko napišemo kratek program, ki nam bo prebiral tipkovnico s pomočjo podprogramov v ROM-u:

```
CD8E02 BERI CALL KEY_SCAN
200F JR NZ, NIC
CD1E03 CALL KEY_TEST
300A JR NC, NIC
5F LD E, A
```