

## LPRS asembler skup instrukcija

| Operacija             | C funkcionalnost              | Kod instrukcije | Komentar   |
|-----------------------|-------------------------------|-----------------|--|
| <b>mov</b> Rz, Rx     | Rz = Rx                       | 00 0000         | Prebacuje sadržaj registra X u registar Z                            |
| <b>add</b> Rz, Rx, Ry | Rz = Rx + Ry                  | 00 0001         | Zbir registara X i Y upisuje u odredišni registar Z                  |
| <b>sub</b> Rz, Rx, Ry | Rz = Rx – Ry                  | 00 0010         | Razliku registara X i Y upisuje u odredišni registar Z               |
| <b>and</b> Rz, Rx, Ry | Rz = Rx & Ry                  | 00 0011         | Rezultat logičkog i registara X i Y upisuje u registar Z             |
| <b>or</b> Rz, Rx, Ry  | Rz = Rx   Ry                  | 00 0100         | Rezultat logičkog ili registara X i Y upisuje u registar Z           |
| <b>not</b> Rz, Rx     | Rz = ~ Rx                     | 00 0101         | Negiranu vrednost registra X upisuje u odredišni registar Z          |
| <b>inc</b> Rz, Rx     | Rz = Rx + 1                   | 00 0110         | Inkrementira registar X i upisuje u odredišni registar Z             |
| <b>dec</b> Rz, Rx     | Rz = Rx – 1                   | 00 0111         | Dekrementira registar X i upisuje u odredišni registar Z             |
| <b>shl</b> Rz, Rx     | Rz = (unsigned short) Rx << 1 | 00 1000         | Logički pomeren registar X uлево upisuje u odredišni registar Z      |
| <b>shr</b> Rz, Rx     | Rz = (unsigned short) Rx >> 1 | 00 1001         | Logički pomeren registar X uдесно upisuje u registar Z               |
| <b>ashl</b> Rz, Rx    | Rz = (signed short) Rx << 1   | 00 1010         | Aritmetički pomeren registar X uлево upisuje u odredišni registar Z  |
| <b>ashr</b> Rz, Rx    | Rz = (signed short) Rx >> 1   | 00 1011         | Aritmetički pomeren registar X uдесно upisuje u odredišni registar Z |
| <b>ld</b> Rz, Ry      | Rz = *Ry                      | 10 0000         | Upisuje u Rz sadržaj koji se nalazi u memoriji na adresi u Ry        |
| <b>st</b> Rx, Ry      | *Ry = Rx                      | 11 0000         | Upisuje u memoriju na adresu koja piše u Ry sadržaj iz Rx            |
| <b>jmp</b> ADDR       | jmp                           | 01 0000         | Bezuslovni skok na labelu ADDR                                       |
| <b>jmpz</b> ADDR      | if Z = 1 jmp                  | 01 0001         | Uslovni skok na labelu ADDR ukoliko je rezultat 0                    |
| <b>jmps</b> ADDR      | if S = 1 jmp                  | 01 0010         | Uslovni skok na labelu ADDR ukoliko je rezultat negativan            |
| <b>jmpc</b> ADDR      | if C = 1 jmp                  | 01 0011         | Uslovni skok na labelu ADDR ukoliko je došlo do prenosa              |
| <b>jmpnz</b> ADDR     | if Z != 1 jmp                 | 01 0101         | Uslovni skok na labelu ADDR ukoliko rezultat nije 0                  |
| <b>jmpns</b> ADDR     | if S != 1 jmp                 | 01 0110         | Uslovni skok na labelu ADDR ukoliko rezultat nije negativan          |
| <b>jmpnc</b> ADDR     | if C != 1 jmp                 | 01 0111         | Uslovni skok na labelu ADDR ukoliko nije došlo do prenosa            |