DATA STORAGE

```
04DA   0000     Unpacked character bit pattern
  DC   0000     Unpacked character bit pattern
  DE   0000     Unpacked character bit pattern
  E0   0000     Unpacked character bit pattern
  E2   0000     Holds address for CHIP-8 subroutines
                   (patterns for DXYN instruction)
  E4   0000     Holds address for CHIP-8 subroutines
  E6   0000     Holds address for machine language subroutines
  E8   C0C0     Grid mark pattern
  EA   C0C0     Grid mark pattern
  EC   0005     V0 V1
  EE   0000     V2 V3
  F0   0500     V4 V5
  F2   0000     V6 V7
  F4   1C27     V8 V9
  F6   1C12     VA VB
  F8   0304     VC VD
  FA   0000     VE VF
  FC   80C0     Line pattern (cursor, sample character point)
  FE   FE00     Line pattern for grid
```

M A C H I N E L A N G U A G E S U B R O U T I N E S

UNPACK CHARACTER BIT PATTERN

```
0500  F8  LDI         ;Load R6 with
  01  F6              ;Address of CHIP-8 V6 variable
  02  A6  PLO   R6    ;R6 points to V6
  03  46  LDA   R6    ;Get V6 value - first ASCII code digit
  04  FE  SHL         ;Shift left, moving
  05  FE  SHL         ;LSB's to MSB position
  06  FE  SHL         ; "        "      "
  07  FE  SHL         ; "        "      "
  08  E2  SEX   R2    ;R2 is Stack Pointer
  09  22  DEC   R2    ;Stack Pointer to free location
  0A  52  STR   R2    ;Push - Stack
  0B  06  LDN   R6    ;Get V7 value - second ASCII code digit
  0C  FA  ANI         ;And "AND" it with
  0D  0F              ;0F for LSB's
  0E  F1  OR          ;Then "OR" with Top of stack - packs ASCII code
  0F  FE  SHL         ;Multiply
0510  FE  SHL         ;by 04
  11  AC  PLO   RC    ;RC indexes bit pattern
  12  F8  LDI         ;
  13  06              ;Base address (high byte) character set
  14  7C  ADCI        ;Add carry, if any, from Multiply Instruction
```