```
      GameBoy Assembly Language Commands, 27-Mar-98
      ---------------------------------------------

  Since books on the Z80 are getting harder & harder to find,
 hopefully the information here might be helpful to those
 trying to understand assembly language specific to GameBoy.

  If 'Flags affected' is not given for a command then none
 are affected.


 Commands
 --------

  ADC A,n        - Add n + Carry flag to A.

         n = A,B,C,D,E,H,L,(HL),#
 ld
         Flags affected:
             Z - Set if result is zero.
             N - Reset.
             H - Set if carry from bit 3.
             C - Set if carry from bit 7.


  ADD A,n        - Add n to A.

         n = A,B,C,D,E,H,L,(HL),#

         Flags affected:
             Z - Set if result is zero.
             N - Reset.
             H - Set if carry from bit 3.
             C - Set if carry from bit 7.


  ADD HL,n       - Add n to HL.

         n = BC,DE,HL

         Flags affected:
             Z - Not affected.
             N - Reset.
             H - Set if carry from bit 11.
             C - Set if carry from bit 15.


  ADD SP,n       - Add n to Stack Pointer (SP).

         n = one byte signed immediate value.

         Flags affected:
             Z - Reset.
             N - Reset.
             H - Set or reset according to operation.
             C - Set or reset according to operation.


  AND n          - Logically AND n with A, result in A.

         n = A,B,C,D,E,H,L,(HL),#

         Flags affected:
             Z - Set if result is zero.
```

```
            N - Reset.
            H - Set.
            C - Reset.


   BIT b,r        - Test bit b in register r.

          b = 0 - 7, r = A,B,C,D,E,H,L,(HL)

          Flags affected:
              Z - Set if bit b of register r is 0.
              N - Reset.
              H - Set.
              C - Not affected.


   CALL n         - Push address of next instruction onto
                    stack and then jump to address n.


   CALL cc,n      - Call address n if following condition
                    is true:

          cc = NZ, Call if Z flag is reset.
          cc = Z,  Call if Z flag is set.
          cc = NC, Call if C flag is reset.
          cc = C,  Call if C flag is set.


   CCF            - Complement carry flag.

          If C flag is set, then reset it.
          If C flag is reset, then set it.


          Flags affected:
              Z - Not affected.
              N - Reset.
              H - Reset.
              C - Complemented.


   CP n           - Compare A with n.

          This is basically an A - n subtraction
          instruction but the results are thrown away.

          n = A,B,C,D,E,H,L,(HL),#

          Flags affected:
              Z - Set if result is zero. (Set if A = n.)
              N - Set.
              H - Set if no borrow from bit 4.
              C - Set for no borrow. (Set if A < n.)


   CPL            - Complement A register. (Flip all bits.)

          Flags affected:
              Z - Not affected.
              N - Set.
              H - Set.
              C - Not affected.
```

```
DAA            - Decimal adjust register A.

       This instruction adjusts register A so that the
       correct representation of Binary Coded Decimal
       (BCD) is obtained.

       Flags affected:
           Z - Set if register A is zero.
           N - Not affected.
           H - Reset.
           C - Set or reset according to operation.


DEC n          - Decrement register n.

       n = A,B,C,D,E,H,L,(HL)

       Flags affected:
           Z - Set if reselt is zero.
           N - Set.
           H - Set if no borrow from bit 4.
           C - Not affected.


DEC nn         - Decrement register nn.

       nn = BC,DE,HL,SP

       Flags affected:
           None


DI             - Disable interrupts.

       Flags affected:
           None.


EI             - Enable interrupts.

       This intruction enables interrupts but not immediately.
       Interrupts are enabled after instruction after EI is
       executed.

       Flags affected:
           None.


INC n          - Increment register n.

       n = A,B,C,D,E,H,L,(HL)

       Flags affected:
           Z - Set if result is zero.
           N - Reset.
           H - Set if carry from bit 3.
           C - Not affected.


INC nn         - Increment register nn.

       nn = BC,DE,HL,SP

       Flags affected:
           None.
```

```
  JP n           - Jump to address n.

         n = two byte immediate value. (LS byte first.)


  JP cc,n        - Jump to address n if following condition
                   is true:

         n = two byte immediate value. (LS byte first.)

         cc = NZ, Jump if Z flag is reset.
         cc = Z,  Jump if Z flag is set.
         cc = NC, Jump if C flag is reset.
         cc = C,  Jump if C flag is set.


  JP (HL)        - Jump to address contained in HL.


  JR n           - Add n to current address and jump to it.

         n = one byte signed immediate value


  JR cc,n        - If following condition is true then
                   add n to current address and jump to it:

         n = one byte signed immediate value

         cc = NZ, Jump if Z flag is reset.
         cc = Z,  Jump if Z flag is set.
         cc = NC, Jump if C flag is reset.
         cc = C,  Jump if C flag is set.


  HALT           - Power down CPU until an interrupt occurs.


  LD A,n         - Put value n into A.

         n = A,B,C,D,E,H,L,(BC),(DE),(HL),(nnnn),#


  LD n,A         - Put value A into n.

         n = A,B,C,D,E,H,L,(BC),(DE),(HL),(nnnn)


  LD A,(C)       - Put value at address $FF00 + register C into A.


  LD A,(HL+)     - Same as LD A,(HLI).


  LD A,(HL-)     - Same as LD A,(HLD).


  LD A,(HLI)     - Put value at address HL into A. Increment HL.


  LD A,(HLD)     - Put value at address HL into A. Decrement HL.
```

```
LD (C),A       - Put A into address $FF00 + register C.


LD (HL+),A     - Same as LD (HLI),A.


LD (HL-),A     - Same as LD (HLD),A.


LD (HLI),A     - Put A into memory address HL. Increment HL.


LD (HLD),A     - Put A into memory address HL. Decrement HL.


LD r1,r2       - Put value r2 into r1.

       r1,r2 = A,B,C,D,E,H,L,(HL)


LD n,nn        - Put value nn into n.

       n = BC,DE,HL,SP
       nn = 16 bit immediate value


LD HL,(SP+n)   - Same as LDHL SP,n.


LD SP,HL       - Put HL into Stack Pointer (SP).


LD (n),SP      - Put Stack Pointer (SP) at address n.

       n = two byte immediate address.


LDD A,(HL)     - Same as LD A,(HLD).


LDD (HL),A     - Same as LD (HLD),A.


LDH (n),A      - Put A into memory address $FF00+n.

       n = one byte immediate value.


LDH A,(n)      - Put memory address $FF00+n into A.

       n = one byte immediate value.


LDHL SP,n      - Put SP + n into HL.

       n = one byte signed immediate value.

       Flags affected:
           Z - Reset.
           N - Reset.
           H - Set or reset according to operation.
           C - Set or reset according to operation.


LDI A,(HL)     - Same as LD A,(HLI).
LDI (HL),A     - Same as LD (HLI),A.


NOP            - No operation.


OR n           - Logical OR n with register A, result in A.

       n = A,B,C,D,E,H,L,(HL),#

       Flags affected:
           Z - Set if result is zero.
           N - Reset.
           H - Reset.
           C - Reset.


POP nn         - Pop two bytes off stack into register pair nn.
                 Increment Stack Pointer (SP) twice.

       nn = AF,BC,DE,HL


PUSH nn        - Push register pair nn onto stack.
                 Decrement Stack Pointer (SP) twice.

       nn = AF,BC,DE,HL


RES b,r        - Reset bit b in register r.

       b = 0 - 7, r = A,B,C,D,E,H,L,(HL)

       Flags affected:
           None.


RET            - Pop two bytes from stack & jump to that address.


RET cc         - Return if following condition is true:

       cc = NZ, Return if Z flag is reset.
       cc = Z,  Return if Z flag is set.
       cc = NC, Return if C flag is reset.
       cc = C,  Return if C flag is set.


RETI           - Pop two bytes from stack & jump to that address
                 then enable interrupts.

RL n           - Rotate n left through Carry flag.

       n = A,B,C,D,E,H,L,(HL)

       Flags affected:
           Z - Set if result is zero.
           N - Reset.
           H - Reset.
           C - Contains old bit 7 data.


RLC n          - Rotate n left. Old bit 7 to Carry flag.
```

```
         n = A,B,C,D,E,H,L,(HL)

         Flags affected:
             Z - Set if result is zero.
             N - Reset.
             H - Reset.
             C - Contains old bit 7 data.


  RR n           - Rotate n right through Carry flag.

         n = A,B,C,D,E,H,L,(HL)

         Flags affected:
             Z - Set if result is zero.
             N - Reset.
             H - Reset.
             C - Contains old bit 0 data.


  RRC n          - Rotate n right. Old bit 0 to Carry flag.

         n = A,B,C,D,E,H,L,(HL)

         Flags affected:
             Z - Set if result is zero.
             N - Reset.
             H - Reset.
             C - Contains old bit 0 data.


  RST n          - Push present address onto stack.
                   Jump to address $0000 + n.

         n = $00,$08,$10,$18,$20,$28,$30,$38


  SBC A,n        - Subtract n + Carry flag from A.

         n = A,B,C,D,E,H,L,(HL),#

         Flags affected:
             Z - Set if result is zero.
             N - Set.
             H - Set if no borrow from bit 4.
             C - Set if no borrow.


  SCF            - Set Carry flag.

         Flags affected:
             Z - Not affected.
             N - Reset.
             H - Reset.
             C - Set.


  SET b,r        - Set bit b in register r.

         b = 0 - 7, r = A,B,C,D,E,H,L,(HL)

         Flags affected:
             None.
```

```
SLA n          - Shift n left into Carry. LSB of n set to 0.

       n = A,B,C,D,E,H,L,(HL)

       Flags affected:
           Z - Set if result is zero.
           N - Reset.
           H - Reset.
           C - Contains old bit 7 data.


SRA n          - Shift n right into Carry. MSB doesn't change.

       n = A,B,C,D,E,H,L,(HL)

       Flags affected:
           Z - Set if result is zero.
           N - Reset.
           H - Reset.
           C - Contains old bit 0 data.


SRL n          - Shift n right into Carry. MSB set to 0.

       n = A,B,C,D,E,H,L,(HL)

       Flags affected:
           Z - Set if result is zero.
           N - Reset.
           H - Reset.
           C - Contains old bit 0 data.

STOP           - ???


SUB n          - Subtract n from A.

       n = A,B,C,D,E,H,L,(HL),#

       Flags affected:
           Z - Set if result is zero.
           N - Set.
           H - Set if no borrow from bit 4.
           C - Set if no borrow.


SWAP n         - Swap upper & lower bits of n.

       n = A,B,C,D,E,H,L,(HL)

       Flags affected:
           Z - Set if result is zero.
           N - Reset.
           H - Reset.
           C - Reset.


XOR n          - Logical exclusive OR n with
                 register A, result in A.

       n = A,B,C,D,E,H,L,(HL),#

       Flags affected:
           Z - Set if result is zero.
```

```
                N - Reset.
                H - Reset.
                C - Reset.
```