MEMORY INTEGRAL SUMMER CORE (MISC)

The MISC is a specialized processing cores who's sole purpose is to add up number in memory. The core is available in two flavours: the ultra cheap MISC1 (or processor taking 1-BCE worth of resources) and the non-so-cheap MISC2 that takes 4-BCE of resources. The instruction sets (that we are interested in) are listed below.

MISC1 / 1-BCE Instructions		MISC2 / 4-BCE Instructions	
`ADD A,X`	; Add regsiter X to accululator A	`ADD Z,X,Y`	; Add regs X and Y and save to register Z
`SET A,value`	; Set reg A to a 32-bit constant value	`SET X,value`	; Set reg X to a 32-bit constant value
`LD A,[X]`	; Load word from mem address addr into A	`LD X,[Y]`	; Load word from mem addr Y into X
`MV X,Y`	; Move value from register Y into X	`LD X,[Y+=4]`	; Load word from mem addr Y into X and increment Y by 4 to next address.
`SWP A,X`	; Swap value of accumulator with reg X	`MV X,Y`	; Move value from regsister Y into regsiter X
`CLR X`	; Set register X to 0	`SWP X,Y`	; Swap value of register Y with X
`J addr`	; Jump to address	`CLR X,Y`	; Set regs X and Y to 0 (X & Y can be same)
		`J addr`	; jump to address
Note: Assume X, Y and Z are placeholders for any registers A,B,C,D,E, or F.			

The MISC2 is able to execute each of its instructions at 2x the speed that MISC1 is able to (i.e. while MISC1 is design around a maximum clock of **800MHz**, whereas the MSC2 supports **1600MHz**).

A typical program that is run on these processors is to sum up important parts of memory to do checksums. A typical program would be:

```
; MISC1 summing procedure
; (don't worry that it doesn't exit)
      CLR B        ; clear the sum register
      SET A,#1000  ; A = mem start address
      MV  F,A      ; F = A
Loop: LD  A,[F]    ; A = mem[F]
      ADD A,B      ; A = A + B
      MV  B,A      ; B = A
      SET A,#4     ; A = 4
      ADD A,F      ; A = A + F
      MV  F,A      ; F = A
      J   Loop     ; repeat the loop
```

```
; MISC2 basic conversion for
; summing procedure
      CLR B
      SET A,#1000
      MV  F,A
Loop: LD  A,[F]
      ADD A,B
      MV  B,A
      SET A,#4
      ADD A,A,F
      MV  F,A
      J   Loop
```

Note: assume that you want the sum to be stored in register B.