OAM DMA Transfer

FF46 - DMA (DMA Transfer and Start Address) (R/W)

Writing to this register launches a DMA transfer from ROM or RAM to OAM (Object Attribute Memory). The written value specifies the transfer source address divided by $10 that is, source and destination are:

```
Source:      $XX00-$XX9F   ;XX = $00 to $DF
Destination: $FE00-$FE9F
```

The transfer takes 160 machine cycles: 152 microseconds in normal speed or 76 microseconds in CGB Double Speed Mode. On DMG, during this time, the CPU can access only HRAM (memory at $FF80-$FFFE); on CGB, the bus used by the source area cannot b used (this isn't understood well at the moment; it's recommended to assume same behavi as DMG). For this reason, the programmer must copy a short procedure into HRAM, and this procedure to start the transfer from inside HRAM, and wait until the transfer has finished:

```
run_dma:
    ld a, HIGH(start address)
    ldh [$FF46], a  ; start DMA transfer (starts right after instruction)
    ld a, 40        ; delay for a total of 4×40 = 160 cycles
.wait
    dec a           ; 1 cycle
    jr nz, .wait    ; 3 cycles
    ret
```

Because sprites are not displayed while an OAM DMA transfer is in progress, most programs execute this procedure from inside their VBlank handler. But it is also possible execute it during display redraw (Modes 2 and 3), allowing to display more than 40 sprite on the screen (that is, for example 40 sprites in the top half, and other 40 sprites in the bottom half of the screen), at the cost of a couple lines that lack sprites due to the fact tha during those couple lines the PPU reads OAM as $FF. Besides, graphic glitches may happ if an OAM DMA transfer is started during Mode 3.

A more compact procedure is

```
run_dma:          ; This part is in ROM
    ld a, HIGH(start address)
    ld bc, $2846  ; B: wait time; C: LOW($FF46)
    jp run_dma_hrampart

run_dma_hrampart:
    ldh [c], a
.wait
    dec b
    jr nz, .wait
    ret
```

This saves 5 bytes of HRAM, but is slightly slower in most cases due to the jump into the HRAM part.