

IPLROM v4.0.1

1. はじめに

IPLROM とは、OCM の起動処理を司る 2KB の Z80 コードである。

SerialROM (EPCS) や SD カード上にある BIOS ROM イメージを読み取って、FPGA ボード上の SDRAM へ転送してから、転送後の BIOS へ飛ぶことにより起動処理を実現している。
本書は、その使い方をまとめたものである。

2. IPLROM v3 からの変更点

(1) BIOS イメージファイルの構造変更

IPLROM v3 までは、BIOS ROM を決まった順序で結合したファイルを BIOS イメージファイルとしていた。

IPLROM4 からは、512byte のヘッダが付いており、そこに記載のコマンドに基づいて、ヘッダより後のデータを読み込み SDRAM へ転送するように変更している。従って、BIOS ROM イメージを作り直す必要がある。

(2) 可変長 BIOS イメージファイル

1 block = 16KB として、次の N block は SDRAM の X 番地に転送せよ、というコマンドを用意している。必要な ROM イメージだけ転送する記述にすれば良いので、必要な ROM が少なければ BIOS イメージファイルを小さく出来て、かつ起動時の読み込みも少なくなる（＝高速になる）。

(3) FFh フィル

未使用の領域を放置には出来ないため、必要の無い領域には FFh を詰めるコマンドを用意している。これも SDRAM の X 番地には N block の FFh フィルをする、という指定になる。

(4) メッセージ表示

BIOS ROM イメージを作成している最中、SerialROM からなのか SD カードからなのか、どちらから起動したか分からないと困る場合や、どのイメージから起動したか分からなくなる場合などがある。

そのような状況をサポートするために、メッセージ表示を搭載している。
もちろん、起動時に何も表示しないように指定することも可能である。

(5) PAL/NTSC 出力の選択

特に MSX1 互換として起動する場合、BIOS 側には PAL/NTSC 選択の設定が存在しないため、PAL モニターを接続している場合に不便である。

(6) MSX1 風カラーパレット初期化

MSX2 の起動時カラーパレットは、MSX1 よりも彩度が高い色合いに変化している。しかし、MSX1 向けに作られたゲームソフト等は MSX1 の色合いでないと見づらい・違和感がある表示になることがある。

この違和感を少しでも和らげるために、MSX1 の色に似せたカラーパレットで初期化する。

MSX2 の BIOS は、通常の MSX2 初期カラーパレットで初期化するため MSX2 の色になる。

MSX1 の BIOS は、カラーパレットレジスタにアクセスしないため MSX1 風パレットで動作する。

(7) I/O ポートへの書き込み

BIOS イメージ上に I/O ポートへの書き込みを埋め込むことが出来る。主に OCM Device に対する各種切替設定の書き込みに利用する。

3. BIOS Image Maker

BIOS イメージを手作業で作るのは大変なので、生成ツールを用意した。

tool\bios_image_maker

Windows のコマンドラインアプリで、第 1 引数に設定ファイル名、第 2 引数に生成する BIOS イメージファイル名を記入する。使い方の一例を表 1BIOS Image Maker の使い方一例に示す。

表 1BIOS Image Maker の使い方一例

| |
|--|
| bios_image_maker.exe ocm_msx2.cfg OCM_MSX2.BIN |
|--|

設定ファイルは、テキストファイルである。

tool\bios_image_maker\ocm_msx2.cfg に、サンプルの設定ファイルを用意している。

設定ファイルは 1 行 1 コマンドであり、; で始まる行はコメントである。空行も無視される。

3.1. メッセージ表示の切り替え

下記の書式でメッセージ表示の切り替えができる。

| |
|-------------------------------|
| DISPLAY_MESSAGE = "ON" |
|-------------------------------|

"ON" は、メッセージ表示する。

これを "OFF" にすると、メッセージを非表示にする。

このコマンドは、全体を通して一番下に記述したものが有効である。

3.2. PAL/NTSC の切り替え

下記の書式で PAL/NTSC の切り替えができる。

| |
|------------------------------|
| MONITOR_TYPE = "NTSC" |
|------------------------------|

"NTSC" は、NTSC 出力想定の設定である。

これを "PAL" にすると、PAL 出力想定に切り替わる。

具体的には、VDP の R#9 の該当ビットに設定する。

DISPLAY_MESSAGE = "ON" の場合のメッセージ表示も、この R#9 に対する設定完了後に行われる。

このコマンドは、全体を通して一番下に記述したものが有効である。

3.3. メッセージ表示

下記の書式でメッセージを表示できる。

| |
|------------------------------------|
| MESSAGE = "message" |
|------------------------------------|

message の部分が表示される。

決まった位置に表示されるため、連続して複数指定しても上書きされて見えないので注意すること。

どの BIOS イメージファイルなのかを識別するメッセージを 1 つ指定するのが通常。

このコマンドは記述順に実行される。

3.4. ROM イメージ読み込み

下記の書式で ROM イメージを読み込む。

| | |
|-----------|---------------------------|
| ROM_IMAGE | = "ROM イメージファイル名", バンク 番号 |
|-----------|---------------------------|

ROM イメージファイル名で指定されたファイルを、バンク番号で示される位置に読み込む。

バンク番号を省略すると、直前の続きになる。

ROM イメージファイルは 16KB の倍数を期待しているが、半端な場合は 16KB の倍数に繰り上げられ、足りない分は FFh で埋められる。

このコマンドは記述順に実行される。

3.5. I/O ポート書き込み

下記の書式で I/O ポートに指定の値を書き込む。

| | |
|---------|-----------|
| OUTPORT | = アドレス, 値 |
|---------|-----------|

このコマンドは記述順に実行される。

例えば、VDP の R#7 に F2h を書きたければ、

OUTPORT=0x99,0xF2

OUTPORT=0x99,0x87

とすれば良い。

3.6. FFh フィル

下記の書式で FFh フィルを実行する。

| | |
|------------|-----------------|
| FILL_DUMMY | = バンク 数, バンク 番号 |
|------------|-----------------|

ROM_IMAGE で FFh が 16KB 詰まったファイルを指定してもいいが、その場合起動時に EPCS や SD カードから 16KB 読み込む処理が入る。

バンク数は 16KB 単位。16KB なら 1, 32KB なら 2 といった具合。

本コマンドでは、読み込まずに LDIR 命令で敷き詰めるので高速である。

このコマンドは記述順に実行される。

3.7. 終了コマンド

下記の書式で BIOS イメージを終了できる。

| | |
|-----------|-------|
| TERMINATE | = サイズ |
|-----------|-------|

サイズは、0 を指定すれば単純な終了を意味する。

数値を指定すると、そのサイズになるまで FFh を詰める。サイズは KB 単位で指定する。

SD カードに BIOS イメージを置く場合、フォーマット直後に最初に書いたファイルという制約がある。しかし、いろいろ切り替えて使いたい場合に毎回フォーマットするのは現実的ではない。

そこで、切り替えて使いたい BIOS イメージファイルをすべて同じサイズにしておき、切替は上書きコピーするようにすれば切り替えられるのである。

そのため、同サイズにするための詰め物である。

IPLROM4 は、この詰め物を読まないため、これによる読み出し速度低下はない。SD カード上の領域を多少無駄遣いするが、切り替えられるメリットの方が大きいと考える人は、このサイズ指定を使えば良い。

このコマンドは記述順に実行される。

3.8. ESERAM 接続メモリの切り替え

SDRAM を 1MB 単位で 32 個に分割してあります。どの 1MB に MegaSDHC を接続させるか選択するための命令です。いわゆるアドレスの上位指定です。0～31 で指定します。

| |
|-------------------------------|
| CHANGE_ESERAM_MEMORY = メモリ ID |
|-------------------------------|