

AVR と Z80 で CP / M の改良 (AKI-80 CP/M) の応用例です。

http://star.gmob.jp/koji/data/VRAM_CRT_C.jpg

久しぶりの更新で生存報告みたいですが、ネタがないわけではなく記事にする気力がないだけです。

さて、手持ち部品の消化といきますか。

表示装置としては AVR ビデオ端末の製作でワンチップマイコンを使った端末形式のものを製作済みですが、今回は CPU から直接アクセスできる VRAM ボードです。仕様は NTSC ビデオ出力で CGROM による白黒キャラクタ表示 40 字 × 25 行です。

ワンチップマイコンだと部品が少なくて作り易かったのですが、自分の作品ではない (Grant さんの改変) し何か物足りず、VRAM と CGROM 外付けの昔ながらの方式で作ってみたいと思いました。配線が大変ですが、まあ何とかできるでしょう。

回路設計

http://star.gmob.jp/koji/data/VRAM_book.jpg

画像を作るタイミングの生成回路は、初期は当然ですが全て汎用ロジックで作られていました。その後 HD46505 等の専用 LSI が登場し、作り易くはなったもののバスの配線があるためそれでも容易ではありません。今ではそれら全てをワンチップマイコンのソフトだけでやってしまうので、すごいことだと思います。

HD46505 を入手してもよいのですが、今回は代替として AVR と Z80 で CP / M の改良の製作後、予備として購入していた CPLD(XC9536XL-VQ44C) で実装しました。

昔の製作本の例では、横文字数が 2 進で切れの良い数 (32 文字とか) になっており最上位ビットをブランキング期間に割り当てています。しかし 40 字表示の MZ-80K とか PET とかはどうやっているのでしょうか？

回路図 (Enri's Home PAGE さん) を見てみると、わざわざラッチとプリセッタブルカウンタを使っているんですね。

これらを CPLD に書いていくんですがマクロセルが全然足りない。規模の大きい CPLD の購入も考えたのですが、そうすると手持ちの部品消化にならないし、なるべく有り合わせの部品だけで作りたい、ということで苦肉の策で CPLD を二個に分けました。タイミングを作る部分と VRAM のカウンタで分けると、間の配線は 5 本のみで一個と大して変わりないですよ ^_^;

CPLD 内のブロック図はこんな感じ。CPU から書込ができる状態では、垂直ブランキング期間以外は WAIT が発生するようにしています。書込みの時のみ WAIT が発生するのが理想ですが間に合わないの読み出しでも発生するようになっちゃってます。

CPLD のピンの割当て、VRAM・CGROM の回路

(部品の選定が変なのは全部有り合わせのものにしているからです)

回路変更【グラフィックも表示】

RAM に 32KB のものを載せているのに 1KB しか使わないのが何か勿体ないような。そこでちょっと気が変わり 320 × 200 ドットのグラフィック表示もできるように変更しました。

変更した回路図

だいが複雑になったように見えますが、CPLD はそのままに IO ポートを一個設けて VRAM の空いたアドレスを全部そちらに接続しただけです。VRAM のアドレスは本来は IO ポートではなく素直に CPU のアドレスの方につなぐべきでしょう。ただこうするとグラフィック表示でも書きっぱなしなら 1KB の空間で済みます。

組立

AKI-80 よりやや大きめの基板でどうにか収まりました。バスの配線は同じところに何本もはんだ付けするのが大変なのでピンを立ててラッピングしています。

http://star.gmob.jp/koji/data/VRAM_BRD01s.jpg 拡大 http://star.gmob.jp/koji/data/VRAM_BRD02s.jpg 拡大

CPU との接続

このボードは読み出しができないので VRAM はメインメモリにかぶせて接続し、読み出す時はメインメモリの方から読むこととなります。CPLD の CS に CPU の MREQ とアドレスデコードを AND したものを入れます。

動作テスト

今回は実験としてアドレスデコードをしない MREQ と IO ポートの 1 ビットの AND を CS に接続しています。こうすると IO ポートで書込可不可を制御し、書込み可の時アドレスのどこに書いても VRAM に書き込まれるという変な仕様になります。なぜこんな変なことをしたかということ、本来なら CP/M のメモリサイズを変更しシステムが使わない領域を作るのですが、適当なところを指定して書き込めるなら TPA 内に指定してしまってもプログラムが小さければ動く可能性が高いからです。

これはいろいろ問題が多いので実験後は普通にした方がいいかなと思います。まず起動時に IO の出力が不定なので、いきなり WAIT が入って起動できなかったり、パラパラ変わる変な画面を見ることとなります。面白いですけどね。

プログラムを書くときも結構嵌ります。書込可の状態では VRAM への書込以外で書込が発生する命令は書けない。例えば PUSH 命令。と、まあ、がんじがらめですw

ソフトウェア

ハードの方は誤配もなくすんなり動いたんですが、ソフト作りは面倒くさくてちっとも進みません。

キャラクタ表示用の回路でのグラフィック表示は、同一アドレスに対し 8 本のラインを割り当てているので、IO ポートでラインを選択して書き込みます。

CP/M 上の HI-TECH C でライブラリを作成し、テストプログラムを動作させてみました。
(ブラウン管は歪むので液晶ディスプレイに表示)

http://star.gmob.jp/koji/data/VRAM_LCD-Cs.jpg1 http://star.gmob.jp/koji/data/VRAM_LCD-G01s.jpg2 http://star.gmob.jp/koji/data/VRAM_LCD-G02s.jpg3
http://star.gmob.jp/koji/data/VRAM_LCD-G03s.jpg4 http://star.gmob.jp/koji/data/VRAM_LCD-G04s.jpg5

- ・昔作りましたね。全部 TTL で、RAM は 1102 だったかな？水晶が高価で OSC は 74123 でしたが、文字は表示できました。 - えびえび (2020 年 10 月 02 日 01 時 52 分 49 秒)