

AVR と Z80 で CP / M の改良の更なる改良の検討です。

http://star.gmob.jp/koji/data/AKICPM_HALT.JPG

別の方法を試す

I/O に WAIT を入れる正攻法的方式では、タイミングが間に合わないのやや複雑なゲートが必要になり CPLD を使いました。できればこれを無くせないかと考えていたのですが、すごいアイデアを考え付く人がいるものです。T colon さんの Tweet。HALT と RESET を使う方法です。実際に PSoC4 で CP/M を動かしていらっやいます。

私が製作した AKI-80 の CP/M マシン (AVR と Z80 で CP / M の改良) ならちょっとの変更で試すことができるのでやってみました。但し HALT からの復帰に RESET を使うと IFF が初期化されてしまい割り込みに影響するので NMI でやることにしました。NMI なら RETN 命令で復帰すれば IFF が保存され INT 割り込みに影響を与えません。CP/M だけなら関係ないですが本物の Z80 を使うからには INT 割り込みは残しておきたいですからね。

回路図

問題なく動きましたが、Z80 側のコードが多くなるので IPL(BOOT) が 1 セクタに収まりきらず 2 セクタ分になりました。この方式の良い点はゲートが不要になるだけでなく I/O アドレスを消費しないところですかね。

詳細はソースを見ていただくとして処理は大体こんな感じ。

Z80

1. コマンドとデータを RAM に書き込む
2. NMI 割り込みベクタ (0066H から 2 バイト分) をバックアップ
3. 割り込みベクタに RETN 命令を書き込む
4. HALT 命令実行
5. バックアップを書き戻す
6. RAM から結果を受け取る

AVR マイコン

1. HALT 検出
2. BUSREQ 発行
3. BUSAK 検出
4. RAM からコマンド、データを取得
5. コマンド処理
6. 結果を RAM に書き込む
7. NMI 発行
8. BUSREQ 解放
9. HALT 脱出 (NMI の受理) を検出

10.NMI 解放

CPLD が不要になっちゃいました。ところでこの PSoC4 って異常に安くないですか。

関連リンク

PSoC4 Prototyping Kit CY8CKIT-049-42XX

<http://akizukidenshi.com/catalog/g/gM-08446/>

<http://itcorp24.cart.fc2.com/ca9/46/>

マイコン PCB [PCB-ATMEGA128](ATmega64/128 対応)

<http://www.aitendo.com/product/2837>

当サイト内

[AVR と Z80 で CP / M の改良](#)

[CP / M コーナー](#)

-
- ・ シンプルで良いですね ~ !、昔の X68K 用の Z80 カードで HALT 使ってありました ~
<https://twitter.com/DragonBallEZ/status/461566441758535682> - kyo (2014 年 06 月 12 日 19 時 57 分 11 秒)
 - ・ 当時からこのテクは使われていたんですね。興味深い情報、ありがとうございます。 - 管理者 (2014 年 06 月 12 日 22 時 52 分 30 秒)