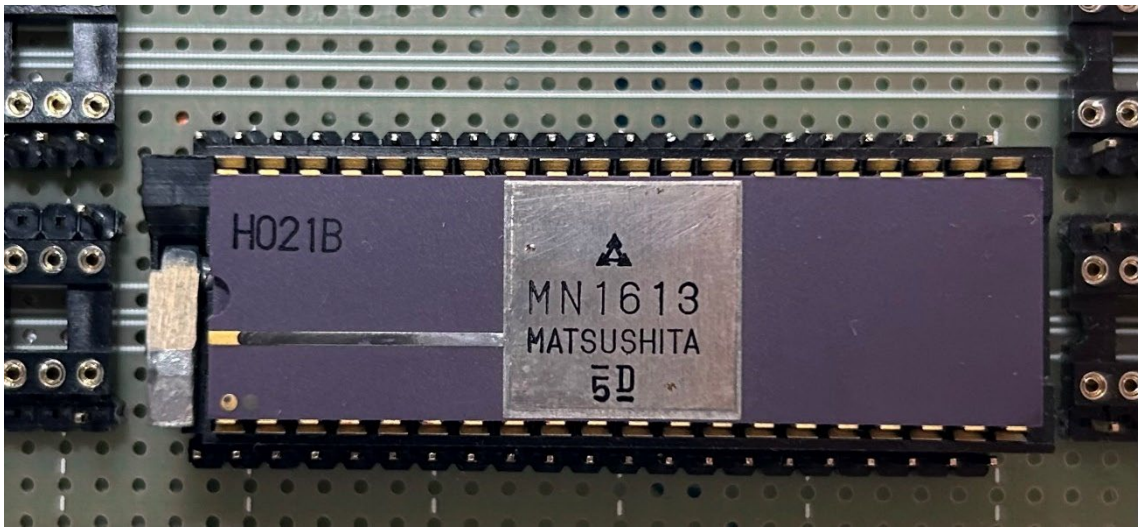
MN1613 マイコンボードの製作

―設計編―



Mercury 日本橋

MUTIF Project

目次

はじめに ..... - 3 -
MN1613 とは ..... - 4 -
　ソフトウェア面から見た特徴 ..... - 4 -
　ハードウェア面から見た特徴 ..... - 7 -
自作 MN1613 マイコンボードの製作 ..... - 11 -
各基板（モジュール）の説明 ..... - 13 -
今後の開発予定・次号予告 ..... - 19 -

はじめに

前作の General Instrument 製の 16 ビットマイコン CP-1600 を使ったブレッドボードマイコンの続編として、今度は、旧パナファコム製 16 ビットマイクロプロセッサ MN1613 を使った「マイコンボード」の製作を試みているので、その途中経過を報告します。CP-1600 ほどではないでしょうが、MN1613 もそれほど知名度が高いとは思えないので、まずは昔語り風に MN1613 の説明から始めたいと思います。

今を遡ること 40 数年、「マイクロコンピュータ」というものが一般化する前の時代。そんな時期にあって秋葉原電気街の一部ではメーカー主導で普及のための拠点が開設され、マイクロプロセッサの評価用・学習用キットの販売が始まっていました。例えば、改装前のラジオ会館の 7 階に Bit-INN[1]ができ、ラジオデパートには、そうした学習キットの広告が貼られるなどしていました。とはいえ価格的には（ミニコンピュータ本体や、その周辺機器と比較すれば遥かに安価だったのだろうとは思いますが）決して個人が手軽に手を出せるレベルではありませんでした（当時既に流行し始めていた BCL[2]用の短波ラジオの数倍といったところ。物価を考慮せず単純に価格で言うと、現在のスマートフォンやタブレットの上位機種の最上位モデルを一括払いで購入するような感じ）。それでも、当時の電子工作キッズはそれらを見て、よくわからないながらも、マイクロコンピュータとは格好いい、すごそうな、未来的なものだ、という印象を強く受け、大いに興味をそそられ、憧れたものでした。

当時の（マイクロプロセッサの）評価用・学習用キットは今の PC とは大きく異なり、むき出しのプリント基板に数値入力やメモリ内容の表示・書き込み、プログラム実行などのための 5×5 ほどのキーマトリクス [3]、6〜8 桁の 7 セグメント表示器という形態が定番 [4]でしたが、その中で異彩を放っていたのが、パナファコムの Lkit-16 でした [5]。まず外見上特徴的なのが、40+8 個に及ぶキー。これらを使って、1 パスのアセンブラでプログラムを入力することができたのです。命令コード表を見ながらハンドアセンブルして命令コードの 16 進数を入力していく他の機種とくらべて、なんと高級感があったことか [6]！そして機能面として特筆すべきは、なんといっても、独自開発された日本初の 1 チップ 16 ビットマイクロプロセッサである MN1610 を搭載していたことです。この MN1610 の後継として発表（1981

---

[1] ラジオ会館 7 階には、日立のマイコンセンターとかもありました。

[2] BroadCast Listening/Listeners の略。主として海外短波放送を受信・聴取して楽しむ趣味。放送局に受信報告書を送ると、放送局から返礼として受信確認証（ベリカード Verification card）が貰えたため、こちらの蒐集も魅力のひとつでした。

[3] いわゆる「16 進キー」

[4] 今の Raspberry Pi や Arduino をひたすら大きくして、キーと 7 セグメント LED を付けたものを想像すれば大きく間違ってはいません。

[5] 2010 年度の情報処理技術遺産です。https://museum.ipsj.or.jp/heritage/PANAFACOM_Lkit-16.html

[6] http://www.st.rim.or.jp/~nkomatsu/evakit/LKit16.html

年）されたのが、今回製作するマイコンボードの主役である MN1613 です（設計はパナファコムだったのでしょうが、製造はどうやら松下だった模様です）。

MN1613 とは

現時点で振り返ってみるに、MN1610 と、その後継である MN1613 は広く普及したとはお世辞にも言えませんが、それでも先人の努力と先見の明により、マイコンボードの自作や、プログラミングに必要な情報は現在でも入手可能です。ここでは、まず MN1613 の元となった MN1610 の特徴について述べ、必要に応じて MN1613 が MN1610 と異なる点について補足することにします。

ソフトウェア面から見た特徴

1. アドレス空間

MN1610 のメモリアドレス空間は 16 bit 幅あり、16 bit で 1 word です。従って、直接アクセスできるメモリは最大で 64 K word となります。バイト（8 bit）単位でのアクセスはできません。MN1613 では後述するアドレス拡張用の専用レジスタが追加され、このレジスタの値と併用することにより、メモリアドレス空間は最終的に 18 bit 幅、つまり最大 256 K word に拡張されます。また、メモリとは別に入出力のための外部機器とのデータの読み書きのために I/O 空間があり、I/O 空間は 8 bit 幅になります[7]。

MN1610 の命令は 1 語 1 word です。そのため分岐命令のジャンプ先のアドレス計算が簡単にでき、このことが MN1610 の命令数の少なさ[8]と相俟って、Lkit-16 のモニタがデフォルトで 1 パスアセンブラをサポートしていた（できた）理由と考えられます。

2. レジスタ構成

MN1610 のレジスタは R0～R4 の 5 個の汎用レジスタと、スタックポインタ（SP）、プログラムカウンタ（命令カウンタ、IC）、そしてステータスレジスタ（STR）という特殊レジスタの組み合わせになっています。全て 16 bit 幅で、このうち、R3 と R4 はインデックスレジスタ X0 と X1 としても使われます。このあたり、ミニコンピュータ PDP-11 の影響を伺わせつつ、（汎用レジスタの一部をインデックスレジスタとして使えるアーキテクチャになっていた）CP-1600 っぽくもあります（尤も、CP-1600 自体が機能縮小版 PDP-11 のような感じなので、そういう印象をもつのも当然なのかもしれませんが）。

MN1613 では、これらのレジスタに加えて、メモリアドレス空間拡張用のセグメントベー

---

[7] 後述するように MN1613 では I/O 空間は 16 bit 幅に拡張されています。

[8] MN1610 の命令数は 33 種類しかありません。ただ、MN1613 では 97 種類に増えています。

スレジスタ、その内容退避用のベース退避レジスタなどのレジスタが追加されています。

3.　アドレッシングモード

MN1610 のアドレッシングモードは合計 8 種類あります。アドレッシングモードと実効アドレスとの関係を下記に示します。MN1610 が発表された当時としては、かなり豊富なアドレッシングモードが用意されていたように思えます。

アドレッシングモード[1]	実効アドレス	例[2]
直接 0～255 番地	命令語の下位 8 bit の値	L[3]　R0, 1
IC 相対+127～-128	プログラムカウンタの値＋命令語の下位 8 bit（この場合のみ符号付）	L　2(IC)
間接	命令語の下位 8 bit の値が示すアドレスのメモリに格納されている値	L　R0, (3)
IC 相対間接	プログラムカウンタの値に命令語の下位 8 bit の値を加えた値が示すアドレスのメモリに格納されている値	L　R0, (4(IC))
インデックス修飾 X0	X0 レジスタの値＋命令語の下位 8 bit の値	L　R0, 5(X0)
インデックス修飾 X1	X1 レジスタの値＋命令語の下位 8 bit の値	L　R0, 6(X1)
間接のインデックス修飾 X0	X0 レジスタの値＋命令語の下位 8 bit の値が示すアドレスのメモリに格納されている値	L　R0, (7)(X0)
間接のインデックス修飾 X1	X1 レジスタの値＋命令語の下位 8 bit の値が示すアドレスのメモリに格納されている値	L　R0, (8)(X1)

1) MN1610 のデータシート記載の用語を記載しました。
2) The Macro Assembler AS のソースツリーの inst1610.asm から引用しました。
3) ロード(Load)命令のニーモニック

これを見てわかる通り、ゼロページ以外のメモリをアクセスするのは少々面倒です。その理由は、アドレス空間は 16 bit 分あるのに、命令語は 1 語で 16 bit、そのうちアドレス指定用には 8 bit 分しか割り当てられていないためです。そのため、メモリの任意の場所を指定するためには、そのアドレス値をどこか適切と思われる場所（その場所をアクセスする命令から-127～+128 の範囲のメモリ）に格納しておいて、IC 相対間接アドレッシングを使ってアクセスすることになります。同じ理由で、インデックス修飾も 8 bit 長の定数インデックスはそのまま使えますが、16 bit 長定数の場合は、インデックス値をゼロページのどこかに格納し、間接インデックス修飾アドレッシングを使う必要があります。このあたりは不便ですが、MN1613 では直接 16 bit のアドレスを指定可能になるなどアドレッシングモードが追加され、便利になっています（2 word 命令が追加されたため）。

4.　条件分岐命令

最も特徴的と思われるのは条件分岐です。MN1610 では演算（比較を含む）命令に、「演算結果次第で次の 1 命令をスキップする」というオプションがあり、これが他のマイクロプロセッサでいう条件分岐命令に相当します。スキップさせないことも可能で、この場合は単純な演算命令ということになります。例を挙げると、

命令	意味
C[1)] R0,R1,Z	R0 = R1 でスキップ
C R0,R1,NZ	R0 ≠ R1 でスキップ
C R0,R1,LPZ	R0 ≥ R1 でスキップ
C R0,R1,LM	R0 < R1 でスキップ

1)　比較(Compare)命令のニーモニック

上の表で言うところの Z とか NZ といったものが「スキップ条件」で、演算結果がこの条件を満たした場合にスキップが実行されます。「スキップ条件」はオプションで省略することができ、省略するとスキップしないという意味になります。

こうした命令スキップをもつ命令がある例としては、ミニコンピュータの PDP-8 が挙げられると思います。他に似たところだと、CASIO のプログラム電卓 FX-502P/602P[9]のプログラムで使われる比較命令（$\boldsymbol{x = 0, x \geq 0, x = F, x \geq F}$）でしょうか（これらは「条件不成立で次の命令をスキップ」でした）。

演算結果によって命令をスキップする動作の特殊例として、IMS 命令と DMS 命令があります。これは、指定したメモリをインクリメント（IMS 命令）あるいはデクリメント（DMS 命令）して、その結果が 0 であれば次の命令をスキップする命令です。これなど、まさに FX-502P/602P の **ISZ**（Increment and Skip on Zero）、**DSZ**（Decrement and Skip on Zero）と同じですね。スキップされる部分に無条件分岐命令（B 命令）を置けば、繰り返しを実現できます。

5.　I/O 空間（I/O ポート）

I/O 空間に接続した入出力機器からのデータ読み出し・書き込みのための専用命令として、RD 命令（I/O 空間から読み出した値をレジスタに格納）と WT 命令（レジスタの値を I/O 空間へのデータ書き込み）が用意されています。前述したように指定可能な I/O 空間は 8 bit 幅です。MN1613 では、これが 16 bit 幅に拡張されています。

6.　その他

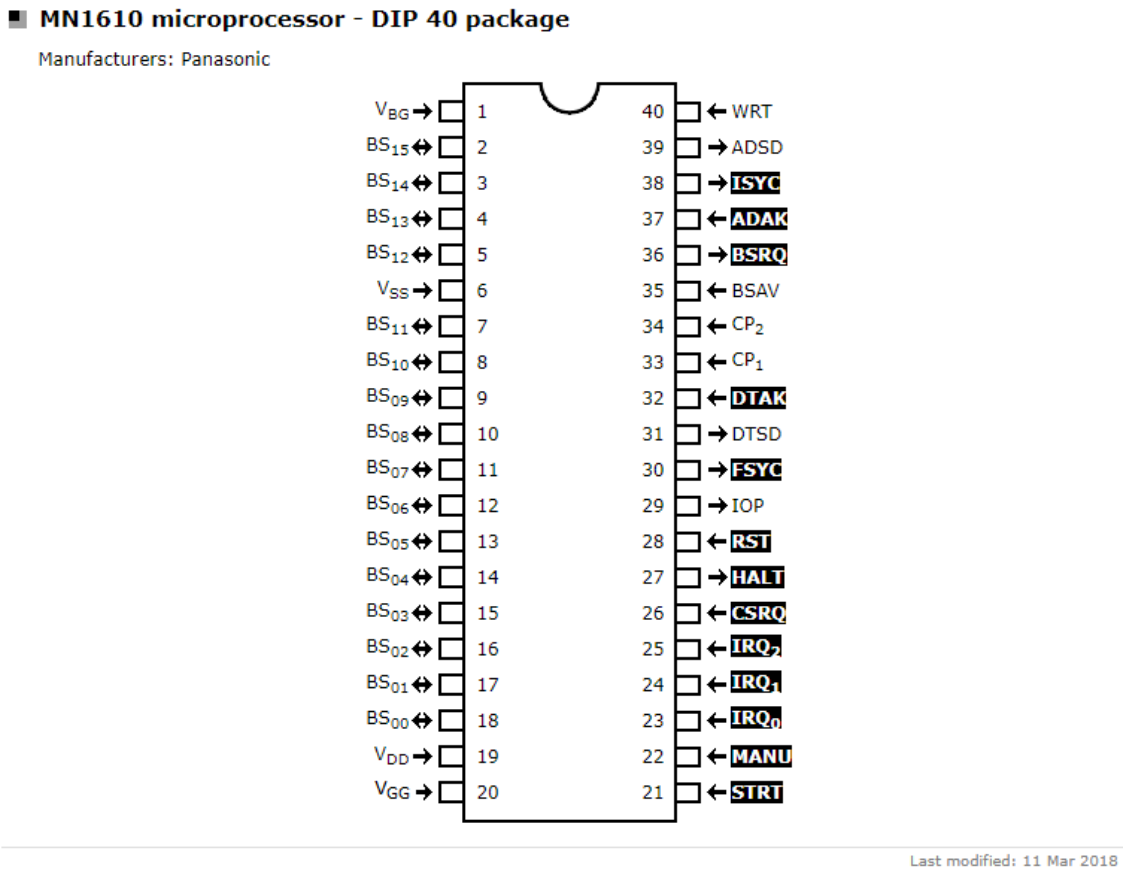
MN1610 には Intel 8086/8088 のような乗除算命令はありませんが、MN1613 になると、符号なしの整数乗算（M 命令）、整数除算命令（D 命令）の他、単精度の浮動小数点演算命

[9] FX-501P、FX-601P も含みます。

令（四則演算命令と、整数と浮動小数点数との相互変換命令）が追加されていることです。数値計算コプロセッサの追加なしに、MN1613 単体で浮動小数点演算ができることは特筆すべきことでしょう。MN1613 には Zilog Z80 や 8086/8088 に似た、ブロック転送命令もあります。

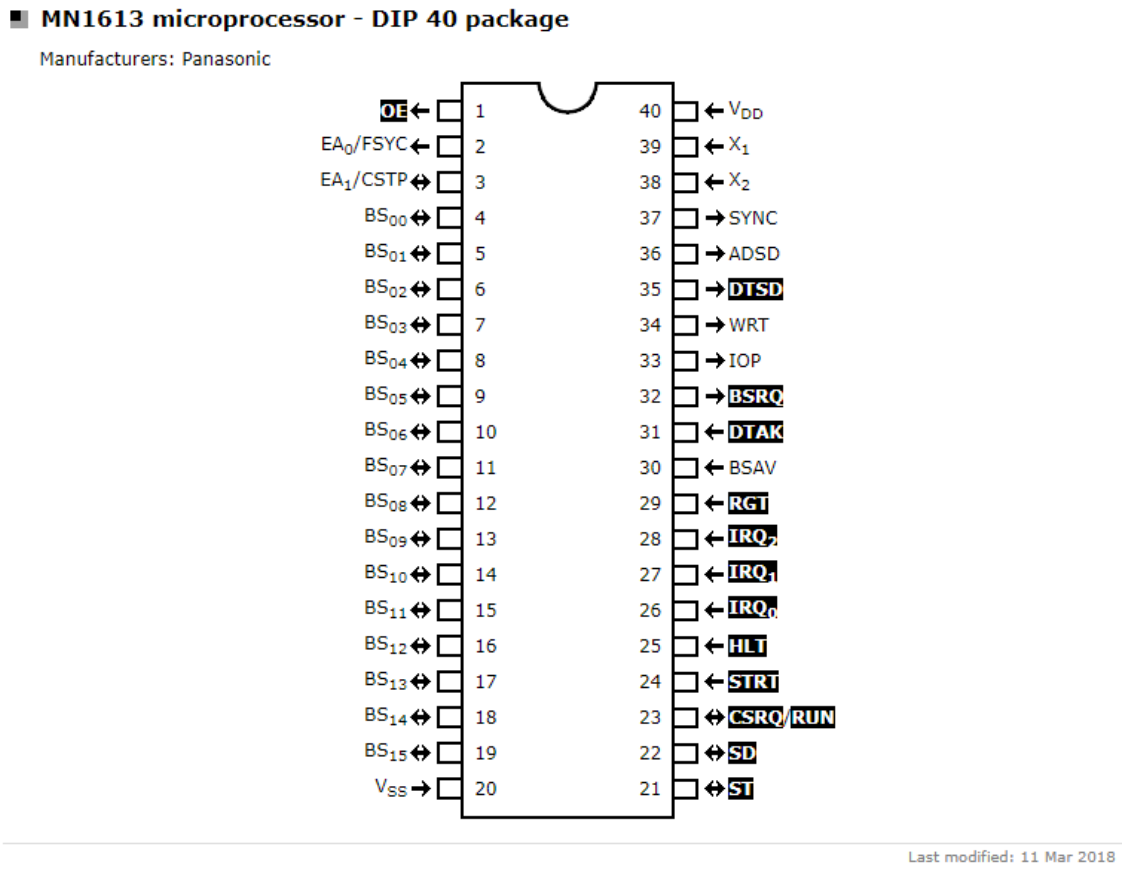
ハードウェア面から見た特徴

まず、MN1610 のピン配置を次に示します（CPU-World より引用[10]、一部改変）。矢印は信号の向きを、黒背景の白字は当該信号が負論理であることを示しています。
パッケージはお馴染みの 600 mil 幅の DIP、40 ピン。BSnn はバス信号で、BS15 が最下位ビット、BS0 が最上位ビットになります。VBG、VDD、VGG、VSS は電源入力です。以下、特徴的な信号について、説明していきます。



[10] https://www.cpu-world.com/info/Pinouts/MN1610.html
なお、サイトに掲載されているものでは 20 番ピンが CTRL となっていましたが、ここでは MN1610 のデータシートに合わせて修正しています。

次の示すのは MN1613 のピン配置です。出典は同じく CPU-World です[11]。



見てお気づきの通り、1 番～20 番ピンの側にバス信号、21 番～40 番ピンの側に各種制御信号という配置こそ踏襲されていますが、MN161 0 と MN1613 ではピン配置がかなり違っているので、単純に置き換えはできません。信号線的に見た MN1610 から MN1613 からの変更点の主なものは、5 V の単一電源になったことに伴って V_{BG}、V_{GG} がなくなったこと、メモリアドレス空間が 16 から 18 bit 幅になったことに対応して、EA0、EA1 が追加されたこと、バス制御信号のうち、$\overline{ADAK}$がなくなったこと、DTSD が負論理になったこと、でしょう。

1.　電源

上述した通り、MN1610 は 3 種類（V_{DD}=5 V、V_{GG}=12 V そして V_{BG}=-3 V）[12]の電源が必要でしたが、MN1613 では 5 V の単一電源（V_{DD}=5 V）で動作します。

[11] https://www.cpu-world.com/info/Pinouts/MN1613.html

[12] V_{SS} = 0 V です。念のため。

2.　クロック

CP1、CP2 ピンがクロック信号の入力ピンです。互いに同時に HIGH になることのない、12 V 振幅の 2 種類のクロックが必要です。データシートを見る限り、最大クロック周波数は 2 MHz で、HIGH 時に流れる電流は最大でも 30 μA なので[13]、おそらくは PIC などのマイクロコントローラとデッドバンド PWM で信号を生成して、それを CD40109 のようなレベル変換 IC を通せば、なんとかなりそうです[14]。当時は周辺 IC として MN1640 というものがあり、これを使えば簡単に必要な 2 相クロックを得ることができました。

MN1613 では 1 相のクロックで済むようになった関係で、クロック入力ピンは X1、X2 という名称になり、ここに水晶発振子を接続するか、外部のクロック信号を X2 に入力すれば済むようになりました。

なお、MN1613 の最大クロック周波数は、13.3 MHz に引き上げられています。

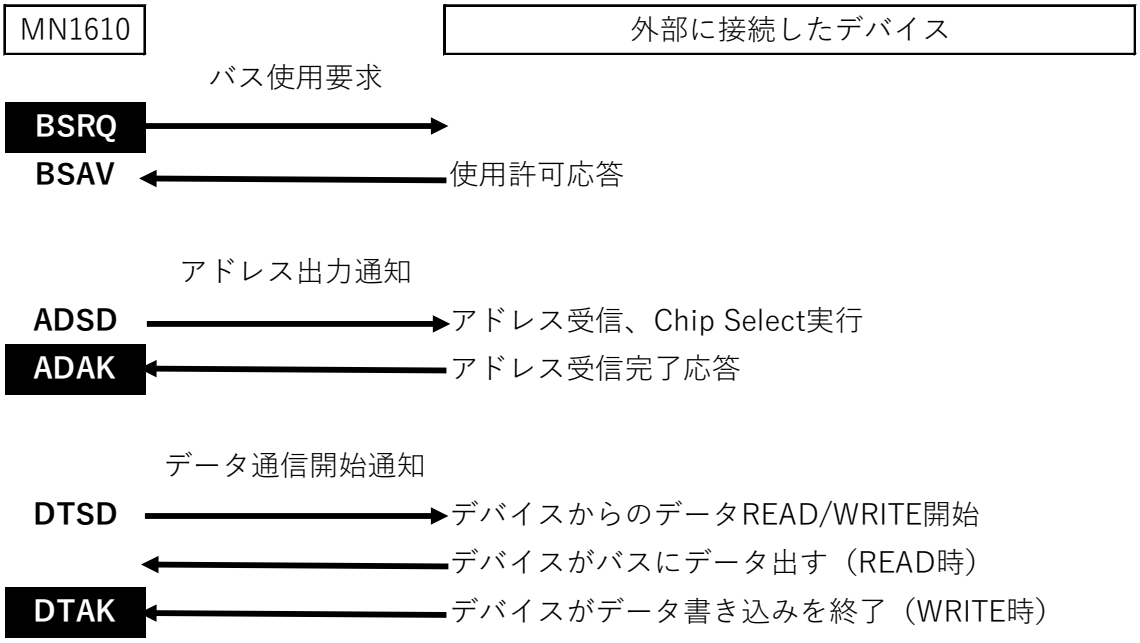
3.　バス信号

ピン配置でいうところの BS0～BS15 がバス信号で、アドレスとデータを同じ端子で時分割して共有します。そして、このバスの現在の役割やデータ入出力の方法を決定するのが $\overline{BSRQ}$、BSAV、ADSD、$\overline{ADAK}$、DTSD および $\overline{DTAK}$ の 6 種類の信号です。$\overline{BSRQ}$、ADSD および DTSD が出力ピンで、これらに対応する入力ピンが、それぞれ BSAV、$\overline{ADAK}$ および $\overline{DTAK}$ です。MN1610 は、これら 3 組 6 種類の信号線によるハンドシェイクを外部デバイスと通信し、データの入出力を行う仕組みになっています。これを示したのが下記の図です。

---

[13] GI 社の CP-1610 のクロックも振幅は 12 V 必要で、そのうえ電流を最大で 15 mA 必要とするため（データシートに記載あり。おそらく CP-1600 も同様）、自作 CP-1600 ブレッドボードマイコンでは 5 V-TTL レベルから必要な電圧・電流を取り出すために SN75365 を使って凌ぎました。

[14] 立ち上がりと立ち下がりの許容時間は最大 50 ns までなので、発振回路の先にオープンドレインのバッファかインバータを接続して出力を 12 V にプルアップといった構成では（少なくとも 74LS シリーズでは）苦しいと思います。



4.　その他

HLT、$\overline{RST}$は他のプロセッサにもあるような動作停止（Halt）、リセット信号です（MN16513 のデータシートには RGT とありますが、おそらくこれは RST の誤記でしょう）。WRT はバスと MN1610 とのデータのやりとりの方向を示し、MN1610 からバスへのデータ出力時に HIGH、バスから MN1610 へのデータ入力時に LOW になります。$\overline{OE}$はこれと似ていますが、WRT と違って負論理です。

IOP は、「ソフトウェア面から見た特徴」で触れた、I/O 空間へのアクセス命令である RD 命令と WT 命令の実行中に HIGH になり、メモリアクセス中は LOW になります。

この他、I/O 空間から 1 語読み込んで、それを命令語として実行するための信号や、未だ役割の良くわかっていない信号もありますが、これらは MN1613 では動作が異なっていたり、信号自体がなかったりするので、MN1613 マイコンボードの製作について解説する本冊子では説明を割愛することにします。詳細については、下記のリンクを参照してください。このような貴重かつ詳細な情報が公開されていることに畏敬の念と感謝を禁じ得ません。

「半導体コレクション展示会場」

MN1610 : 初期 16 bit マイクロプロセッサ

http://www.st.rim.or.jp/~nkomatsu/panafacom/MN1610.html

MN1613 : MN1610 の後継

http://www.st.rim.or.jp/~nkomatsu/panafacom/MN1613.html

自作 MN1613 マイコンボードの製作

MN1613 を使った自作マイコンボードの製作にあたっては、asano（@electrelic）さんが、ご自分の Blog や Twitter で公開されている情報に敬意を払いつつ、大いに参考にさせていただきました。それらの情報なくしては、そもそも MN1613 を使ったマイコンボードを自分でも製作してみようと思いもしなかったでしょう。この場を借りて御礼申し上げます。

今回、製作にあたって最初に考えたことは、できる範囲で回路を機能ごとに分割し、再利用できるようにしよう、ということです。MN1613 の他にも実際に動かしてみたいプロセッサはいくつもあるので、そのたびに 0 から作り直すのは無駄ではないのかと思ったのです。この点でも大いに刺激を受けたのが、asano さんの製作された Emily Board です。これは、様々なプロセッサをとりあえず動かしてみようとした場合、プロセッサは変わっても、最低限必要になるメモリや外部とのやりとりのためのシリアルコンソールといった機能は共通化できるのではないか、というコンセプトで製作された一種の汎用メモリボードで、4 K×8 bit のデュアルポート SRAM と、その片側のポートに接続されたマイクロコントローラ（ATmega）から成ります。マイクロコントローラは、シリアルコンソールの役割とデュアルポート SRAM へのユーザープログラムの書き込み、ターゲットプロセッサ（例えば今回で言えば MN1613）のリセットなどを担当します。反対側のポートはターゲットのプロセッサと接続されます。必要に応じて 4 K×8 bit のデュアルポート SRAM を載せた基板を追加して、8 bit を超えるバスサイズのプロセッサにも対応できるように配慮されたものです。私も MC6809、HD63C09、そして CP-1600 を使った、複数の自作マイコンボードを設計・製作してみて Emily Board のコンセプトに共感するところ多々あり、MN1613 とその周辺 IC（データとアドレスの分離、アドレスのラッチ、クリスタルオシレータなど）は独立した基板とし、それと自分なりの Emily Board を作って MN1613 と組み合わせてみようと思い立ちました。

次に考えたことは、いわゆるグルーロジックのことです。とりあえず動かしてみようというだけでも、プロセッサによってはかなりの数のグルーロジックが必要となります（CP-1600 がそうでした）。かなりの数とはいっても、74 シリーズのロジックゲートを数個組み合わせれば済む規模ではありますが、それでも、それなりに場所を取るうえ、仕様の変更に伴う配線の変更が厄介です（そして、そういった仕様変更は、試行錯誤中には頻繁に起きるものです）。そこで Gate Array Logic などの Programmable Logic Device の登場となるわけですが、昨今の情勢で GAL（Microchip が言うところの SPLD）[15]の入手は極めて困難になっており（納期が年単位）、代替となりそうな、小規模な CPLD は 5 V 品は軒並みディスコ

[15] Lattice Semiconductor の GAL22V10 や、その互換品の Microchip ATF22V10 など

ンになっている[16]か、5 V 品があっても論理合成ツールが有償かつ（個人で開発するには）高価[17]です。そこで、機能的には勿体ないレベルではありますが、近年、比較的容易に入手できるようになった FPGA を使ってみることにしました。

候補としては 2 つ、ひとつは SiPeed 製の Tang Nano 9K という Gowin Semiconductor 製の LittleBee シリーズ FPGA を搭載したボード[18]で、その利点はなんといっても、秋月電子にて 2480 円で購入できる価格と入手性です。そして、論理合成ツールも VS Code を使うフリーのものが利用可能です。難点としては、I/O のピン数は多いものの、複数の機能を同じピンで共有しているものが多く、使い慣れないと思わぬトラブルに遭遇する懸念があります。もうひとつの候補は、EIM Technology 製の STEP FPGA という Lattice Semiconductor 製の MachXO2 シリーズ FPGA[19]を搭載した学習用ボードです。学習用ということで、FPGA の入出力ピンがそのまま引き出されており、（Tang Nano のような）microSD カードや HDMI インターフェイスがない代わり、2 組の 7 セグメント LED を含む多くの LED、DIP スイッチやタクトスイッチのような基本的な部品をオンボードに搭載しており、今回のようなグルーロジックの実装においても、デバッグ用に重宝することでしょう。開発環境も、Lattice Semiconductor 製の純正開発環境があるほか、Web ベースの Verilog コンパイラが使え、書き込みも専用のプログラマを必要としません（ボードを PC と接続すると USB のストレージデバイスとして認識されるので、ビットストリームをコピーするだけ）。こう書くといいことばかりのようですが、大きな問題もあって、それはボードの価格です。1 枚あたり 79.99 カナダドルで、これは本冊子執筆時点での為替レートで日本円に換算すると 9000 円弱になり、これは Tang Nano 9K のおよそ 4 倍弱に及びます。また、入手性もいいとはいえず、メーカーのオンラインストアか、Tindie から通信販売で購入することになります[20]。なお、ロジック数は Tang Nano 9K の約半分といったところでしょうか。

どちらの FPGA ボードも、残念なことに I/O ピンは 5 V トレラントではないので、MN1613 との間にはレベル変換 IC[21]を噛ませることになります。自分としては初めての試みであるのと、物理的なスペースの関係で、MN1613 を載せた基板とは別に、FPGA 基板を独立させることにしました。

このようなわけで、今回の MN1613 マイコンボードは、MN1613 基板、FPGA 基板、そ

---

[16] Xilinx の XC9500 シリーズや ALTERA の EPM7032 など

[17] Microchip の ATF1502/1504/1508

[18] 9K にしたのは秋月電子の店頭で気軽に購入できるからで、他意はありません。

[19] 2022 年 12 月に開催されたコミックマーケット 101 で発表した自作 MachXO2 ボードも今回のような目的を想定したものでしたが、DIP への変換基板のサイズが大きくて今回の使用は見送りました。

[20] https://www.tindie.com/products/evoinmotion/fpga-development-board-step-mxo2/

[21] FPGA の入力ピンと 5 V 系のデバイスとの間に 1 KΩ程度の抵抗をシリーズに挟む、あるいはダイオードを使ってオープンドレイン化して、FPGA 側を 3.3 V にプルアップする、といった方法も考えましたが、波形が鈍ってクロック周波数を上げた時に使えなくなりそうだったので、レベル変換 IC を使う方針に途中で転換しました。

して汎用メモリボード―自分版 Emily Board―Celeste Board と命名しました―の 3 モジュールから構成されます。

各基板（モジュール）の説明

本マイコンボードを構成する 3 つのモジュールについて解説していきます。最初にお断りしておきますが、本冊子の第 1 版を執筆中の 2023 年 7 月 31 日の時点で、マイコンボードは製作中です。そのため、本冊子の本版に掲載されている回路図は、作成後しばらく「寝かせた」後に何度か確認と修正をしてはいますが、実際に動作確認がとれているものではありません。そのことを最初にご承知おきください。なお、回路図や、FPGA に書き込むロジックの Verilog ソースなどの設計情報は、GitHub でも公開・更新予定（下記）ですので、そちらも随時参照いただければと思います。

https://github.com/ja1umi?tab=repositories

1. MN1613 基板

8×D-FF の 74LS574 を 2 つ使って、ADSD で BS00～BS15 に出力されるアドレス値をラッチします。データシートのタイミング図的には ADSD の立ち上がりでも大丈夫そうなのですが、インバータを噛ませて立ち下がりでラッチすることを考えています。

$\overline{DTSD}$、IOP、WRT を 2-to-4 ラインデコーダの 74LS139 でデコードして、メモリ読み書き用の信号を生成しています。I/O 読み書き用の信号も得られますが、現状では未使用です。これは、I/O 関係は MN1613 の I/O 空間を使わずに、後述する Celeste Board 上にある共有 RAM としての Dual-port SRAM を介して行うことにしたためです。

BS00～BS15 の信号線を引き回すのは気が引けたので、$\overline{OE}$を方向制御信号として使って 74LS234 を 2 つでデータのバスバッファとしました。

MN1610 や MN1613 の特徴ともいえるバスのハンドシェイクですが、今回の場合、アイコンボードにバスマスタは MN1613 だけであるのと、データシートに「BSAV=”H”固定可」「$\overline{DTAK}$=”L”固定可」とあったことから、今回は回路を単純にするため、BSAV は 5 V にプルアップ、そして$\overline{DTAK}$は GND にプルダウンしました。

クロック源としては 2 MHz のクリスタルオシレータを使います。MN1613 のクロック周波数の下限は 1 MHz、上限は 13.3 MHz であることを考えると低い値ですが、これはまずは安定した動作するかを確認したかったからで、それが確認できれば徐々に高い周波数のものに交換することを検討しています。

MN1613 のリセットのパルス幅の最小値はデータシートによると 2000 ns とかなり短いので、パルス幅についてあまり深く考えなくてもよさそうでしたが、念のためリセット用の IC である DS1232 を入れました。これはウォッチドッグタイマ機能も持っているのですが、タイマのリセット用の信号には ADSD を使いました。

動作状態を知るためとデバッグのため、にバッファ代わりのインバータを挟んで LED を

接続しています。最近の LED は小電流でも十分に明るいので MN1613 に直結しても問題なさそうでしたが、秋月電子で見かけた正方形の面発光 LED アレイ [22]（内部に 4 本の LED が入っている）を使ってみたかったので、必要な電流を流せるようにとバッファを入れています。貴重な MN1613 を部品削減でダメージを与えるようなことは避けたかったので。$\overline{RUN}$ がアサートされる（RUN 状態になる）と点灯します。

今回特に工夫したのはリセット直後の、プログラムの実行開始アドレス、いわゆるリセットベクタの与え方です。MN1613 はリセット直後に I/O 空間のアドレス 0 から値を読み込んで、その値に 3 を加えたものがリセット後のプログラムのスタートアドレスになることが、既に MN1613 を動かして色々と試されている asano 氏によって明らかにされています [23]。今回はリセット直後の、最初の ADSD の時だけ例外的に EEPROM が選択され、そこから値を読み出すという方法を試してみることにしました。

2. FPGA 基板

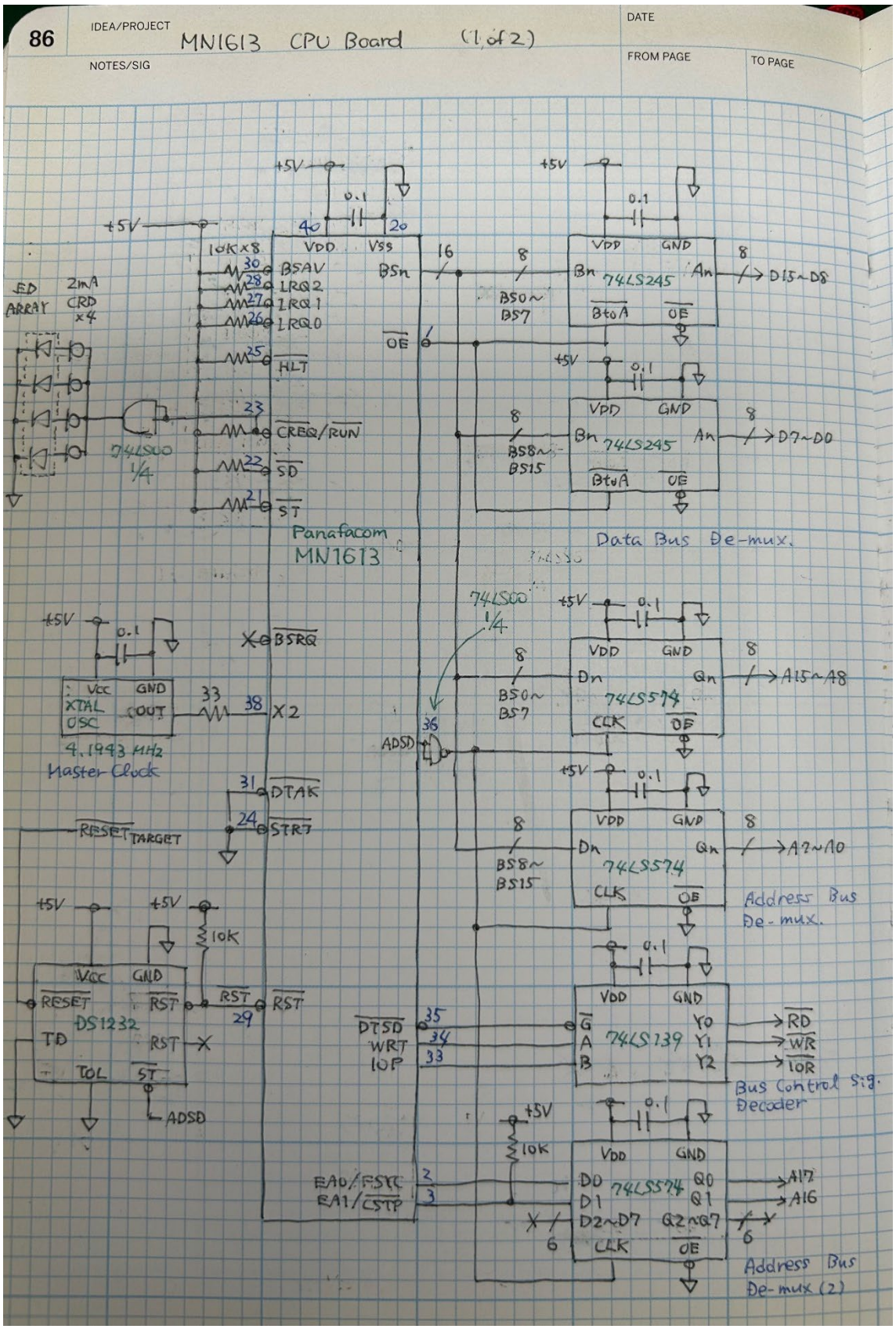
結局 MN1613 の周辺の IC の数が増えたこと、また、使用する FPGA ボードが複数あること、そして 3,3 V と 5 V のレベル変換も必要になることから、FPGA の部分だけ基板を独立させました。

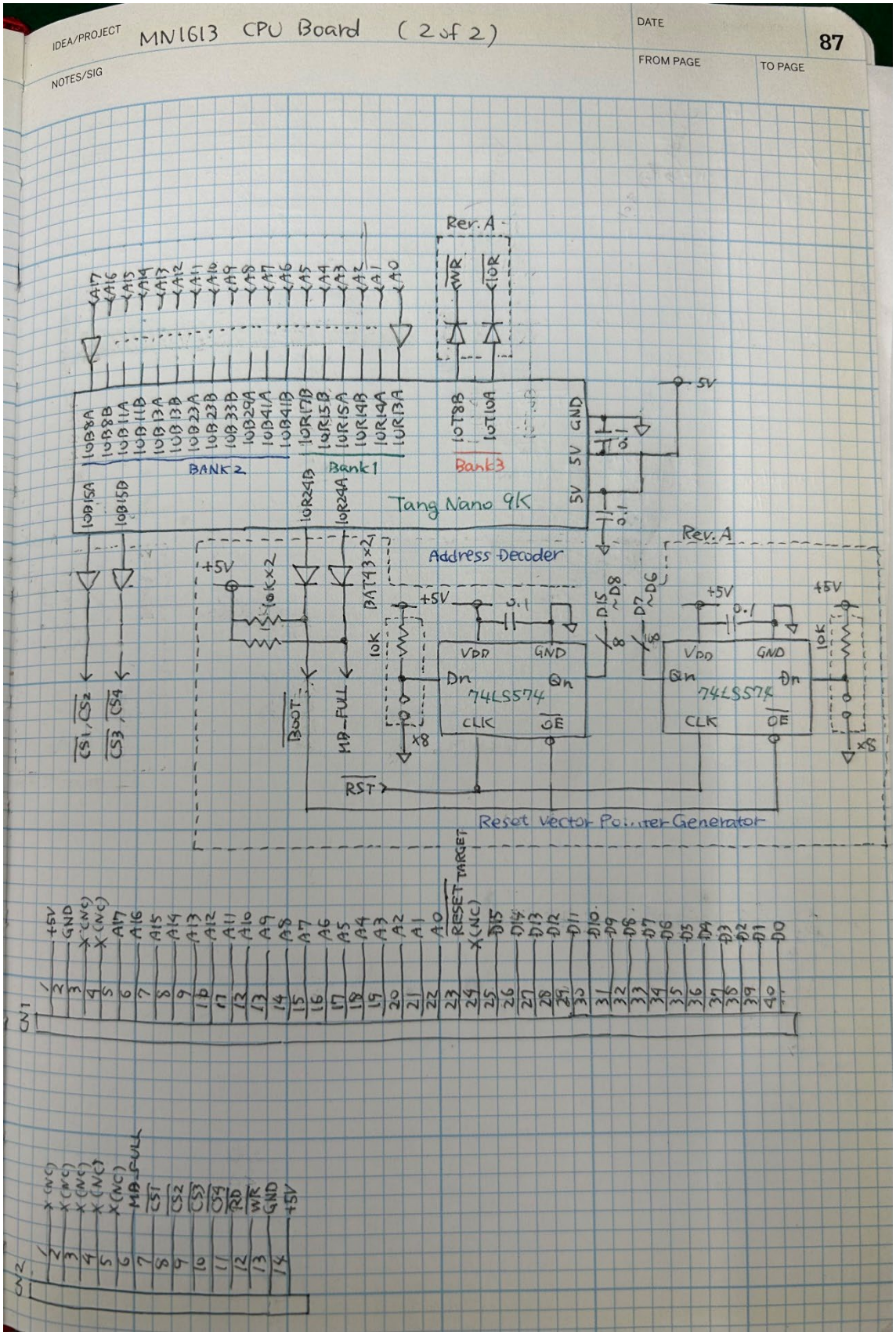
回路としては単純で、単に FPGA の入力ピンのロジックレベルを合わせるため、5 V から 3.3 V へのレベル変換のための SN74CB3T3245 から構成されます。SN74CB3T3245 はバスバッファの 74LS245 と同じ配列の 5V トレラントなアナログスイッチです。アナログスイッチなので、他のレベル変換 IC と比較して遅延がほとんどないので、クロックを上げた時に効いてくるのではないかと期待しています。FPGA からの出力については、フルスイングで 3.3 V 近くまで出るのであれば 5 V ロジックの HIGH レベルと見なされるだろうと楽観的に考えて、レベル変換は特に行っていません。

---

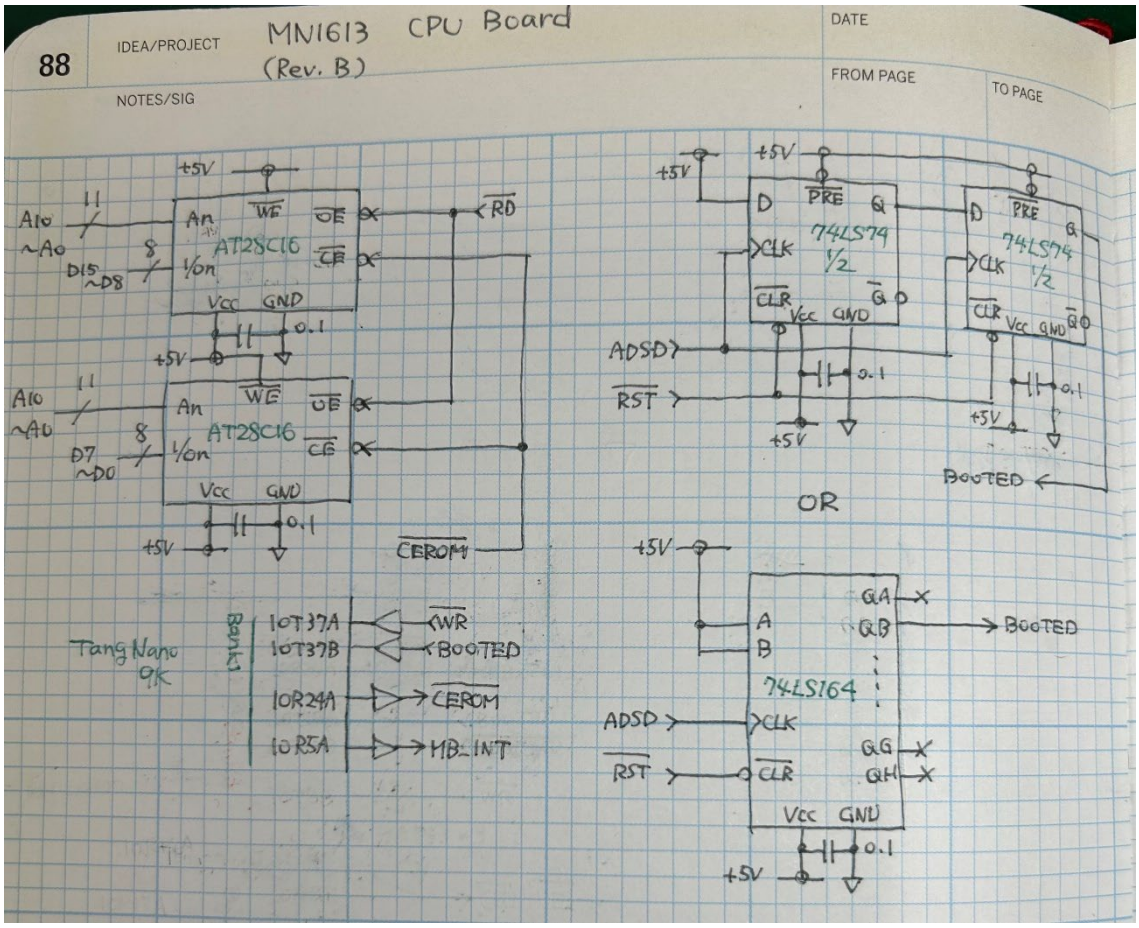
[22] https://akizukidenshi.com/catalog/g/gI-17548/

[23] MN1613 のステップ実行（その 5）　https://electrelic.com/electrelic/node/1609


86
IDEA/PROJECT
MN1613 CPU Board (1 of 2)
NOTES/SIG
DATE
FROM PAGE
TO PAGE
+5V
0.1
40
20
VDD
VSS
10K×8
30 BSAV
28 IRQ2
27 IRQ1
26 IRQ0
25 HLT
23 CREQ/RUN
22 SD
21 ST
LED ARRAY
2mA CRD ×4
74LS00 1/4
Panafacom MN1613
BSn
16
OE
8
BS0~BS7
Bn
74LS245
An
BtoA
OE
D15~D8
BS8~BS15
D7~D0
Data Bus De-mux.
74LS00 1/4
36
ADSD
Dn
Qn
74LS574
CLK
A15~A8
A7~A0
Address Bus De-mux.
X BSRQ
Vcc
GND
XTAL
OSC
COUT
33
38
X2
4.1943 MHz
Master Clock
31 DTAK
24 STRT
RESET TARGET
10K
RST
29
DS1232
TD
TOL
ST
ADSD
DTSD
WRT
IOP
35
34
33
G
A
B
74LS139
Y0
Y1
Y2
RD
WR
IOR
Bus Control Sig. Decoder
EA0/FSTC
EA1/CSTP
3
D0
D1
D2~D7
Q0
Q1
Q2~Q7
A17
A16
6
Address Bus De-mux (2)


IDEA/PROJECT
MN1613 CPU Board (2 of 2)
NOTES/SIG
DATE
FROM PAGE
TO PAGE
87
Rev. A
BANK 2
Bank1
Bank3
Tang Nano 9K
Address Decoder
74LS574
Reset Vector Pointer Generator
RESET TARGET
MB_FULL
CN1
CN2

MN1513 基板の Revision B との変更点は次の通りです。リセット直後のスタートアドレス値をバスに注入する部分と、FPGA へのレベル変換をプルアップ抵抗＋ダイオードの組み合わせではなく、専用の IC（SN74CB3T3245、回路図中ではバッファ（三角記号）として図示）の使用が主な変更部分です。

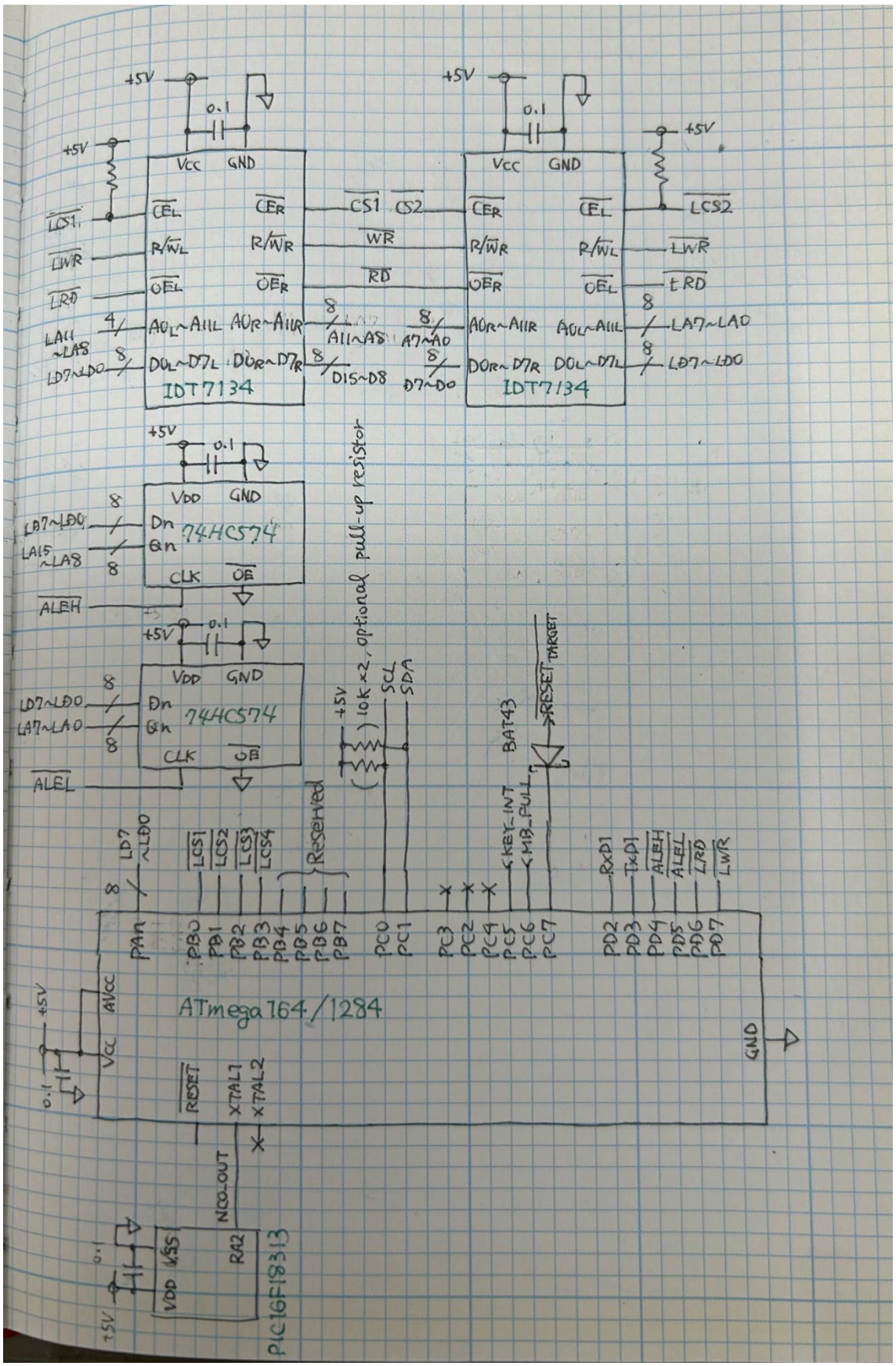


3.　汎用メモリボード Celeste Board

前述した通り、asano さんが発表されている Emily Board を自分なりに手を加えたものです。基本構成は共有メモリとして使うデュアルポート SRAM の IDT7134 と、共有メモリの操作や、シリアルインターフェイスの代替に ATmega164 あるいは ATmega1284 を使います。Emily Board との違いは、ターゲットとなる CPU 基板（今回の例でいうと MN1613 基板）との接続方法とコネクタのピン配置、デバッグ用のキャラクタ液晶や 16 進キーボード、動作チェックのために共有メモリを介して接続する GPIO の類は I2C 接続にした点です。

なお、ATmega164 や ATmega1284 のプログラム開発には、MightyCore[24]を使います。

[24] https://github.com/MCUdude/MightyCore


IDT7134
IDT7134
74HC574
74HC574
ATmega164/1284
PIC16F18313
+5V
0.1
Vcc
GND
VDD
CLK
OE
Dn
Qn
LCS1
LWR
LRD
LA11 ~LA8
LD7~LD0
ALEH
ALEL
CS1
CS2
WR
RD
A11~A8
A7~A0
D15~D8
D7~D0
LCS2
LA7~LA0
10k ×2, optional pull-up resistor
SCL
SDA
BAT43
RESET TARGET
KEY_INT
MB_PULL
RxD1
TxD1
Reserved
AVCC
RESET
XTAL1
XTAL2
NCO_OUT
VSS
RA2

今後の開発予定・次号予告

まずは MN1613 の動作の確認を行う予定です。短いプログラムが動くようになれば、その時点での設計情報の更新分や、その時点までに得られた知見を次号にて紹介する予定です。また、デバッグの便のため、キャラクタ液晶モジュールや 16 進キーボードを使えるように ATmega のファームウェアも開発を進めていきたいところです。

また、MN1613 基板に続いて、Celeste Board で動く CPU 基板として、80286 か 68008 も作ってみたいところですが、まずは MN1613 を動かせてからですね。

表　題　：	**MN1613 マイコンボードの製作　―設計編―**	
サークル名：	Mercury 日本橋	
発 行 者 ：	Kasumi YOSHINO (JA1UMI)	E-mail: ykasumi@spamex.com
印　刷　：	プリントパック様	X/Twitter: @yKasumi
	https://github.com/ja1umi/	MissKey.io: @yKasumi
発 行 日 ：	2023 年 8 月 13 日　コミックマーケット 102（C102）	