

自作のロジックアナライザ（改訂版、Ver.0.4）の概要と取説を示す。

（0）履歴

マイコンのバスラインをモニターする必要が生じたのでロジックアナライザ（略称、ロジアナ）を自作した。自作したのは、DE0-nano という FPGA ボードと、カラーグラフィック液晶表示器が余っていたためである。市販の PC へ USB 接続するロジアナを買う手もあるが、余ったボード類を生かせる他に、自分好みのロジアナを作成できる点、および、USB ロジアナ + PC による机上のスペースを多く取るよりも単独で動作するロジアナの方が使いやすいと思ったためである。（通常、プロは PC 接続が必要な測定器をあまり使うことはなく、測定器単独で動作するものを好む。）

バージョン履歴：

第1バージョン = 0.2、改訂版（波形比較、I2C プロトコル表示を拡張） = ver.0.23。

ver.0.23 の拡張点：SD-Card にストアした binary ファイルと、現在のデータを画像表示で比較できるようにした。I2C プロトコルを解析表示できるようにした。波形取り込みモード下においてもトリガーパラメーターや表示倍率と区間を変更できるようにした。

ver.0.3：サンプリング周波数を増やし、50kHz, 100kHz, 200kHz, 500kHz, 1MHz, 2MHz, 5MHz, 10MHz, 20MHz, 50MHz, 100MHz, 200MHz の選択を可能とした。

（最速 200MHz まで可能。）

ほとんどの操作は ver.0.23 と同様。

ver.0.4：トリガー・スキップ・カウント機能を実装。

SPI プロトコルアナライザ機能を追加（SPI モード 0 のみ）。

トリガー条件に CH0 & CH1 を追加。

ほとんどの操作は ver.0.23 と同様。

（1）概要

ver.0.4 ソフト改訂および FPGA 設計追加 = ver.0.4 の仕様を示す。

構成：ver.0.2 と同じ。DE0-nano board (Color graphic 液晶表示の制御、およびロジアナ信号取得)

AVR Mega1284 基板（ユーザーインターフェース + SD-Card writer）

USB インターフェース（FT232R 仮想 COM ポートにより、PC と通信）

仕様：

液晶表示：800 x 480 dot, 16-bit color (共立デジット、パチスロ液晶)

ロジック波形、および動作状態表示

液晶表示制御：DE0-nano で実現。内蔵 SDRAM で frame buffer。

ロジアナ：DE0-nano FPGA で実現。

sampling frequency = 200 MHz max

channel 数 = 4 channel (FPGA 設計の変更により、任意数)

memory 容量 = 各チャンネルあたり、64k

入力ロジックレベル = 3.3V or 5V logic levels (74LCX244 で入力 buffer)

データストレージ：Mega1284 board で制御

SD-card: Windows format, Text file で波形をファイル書き込み (PC で読み取り可能)

USB: PC へ波形出力、Text format (仮想 COM ポートからファイルへ記録可能)

(2) 取説

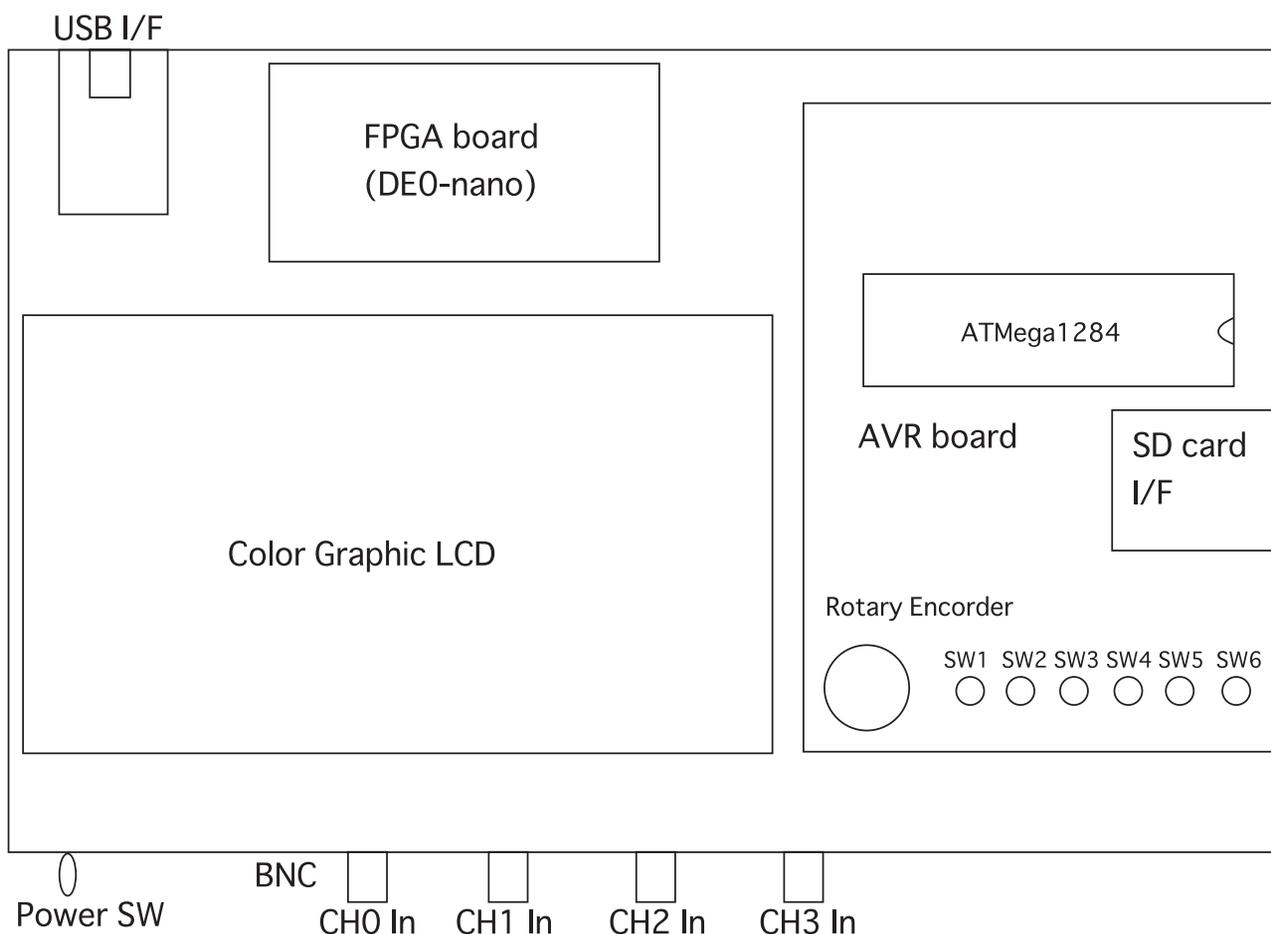


図1 上面図

2-1 パネル面解説 (図1 参照)

BNC connectors : 4つの入力チャンネルがあり、CHO ~ 3 の BNC コネクタより入力

Color Graphic LCD (以後、CGLCD と略) : 波形表示および計測パラメーター等が表示される。

USB I/F : PC と USB ケーブルでつなぎ、HyperTerminal ソフトを利用すると PC との通信ができる。(注 : No flow control で設定すること)

SD card I/F : 波形を SD card にメモリーすることができる。注 : SDHC には対応していない。

Rotary Encoder : 回すとパラメーター等を変更できる。

SW1 ~ 6 : 測定パラメーターの変更や、波形取得の Start / Stop 等の制御に使う。

2-2 モードの設定

SW6 を押すごとに3つのモードが順繰りに切り替わる。

モード0 : ロジアナモード

モード1 : 波形比較とプロトコル表示の設定

モード2 : SD-card モード

モード3 : USB モード

なお、モード2と3は、モード1や2からSW5を押している間に同時にSW6を押さないと切り替わらない。これは、モード2と3は、あまり使わないため、モード0とモード1のみを高速に切り替えるようにしたためである。通常はSW6のプッシュのみでモード0と1の間のみで切り替わり、すみやかに波形取得の設定ができるようにした。(ver.0.2 から ver.0.23 へのソフト改良による。)

2-3 ロジアナモード (モード0)

CGLCD の2行目に LogiAna と表示される。(Power-ON 時の default) このモード下でロジアナの制御を行う。各スイッチは以下の機能を持つ。

SW1 : 波形取得をスタートする。

SW2 : 波形取得をストップする。

SW3 : パラメーター項目選択を進める。

SW4 : パラメーター項目選択を戻す。

SW5 : パラメーター項目選択 / 波形表示エリア選択の切り替え

SW6 : 波形取得条件が満たされない場合に、強制終了をする

各スイッチは以下のように使用する。

まず、トリガー条件を設定する。SW3,4 でパラメーター項目を選択する。以下の項目があり、各々ロータリーエンコーダーで設定値を変えることができる。

ロジアナパラメーター設定状態

Sampling 周波数：50kHz, 100kHz, 200kHz, 500kHz, 1MHz, 2MHz, 5MHz, 10MHz, 20MHz, 50MHz, 100MHz, 200MHz

周波数が高いほど高分解能。max 64k sampling point.

CH0 trigger condition：CH0 でこの trigger condition があれば、波形記録をスタートする。

CH1 trigger condition：CH1 の trigger condition。

CH2 trigger condition：CH2 の trigger condition。

CH3 trigger condition：CH3 の trigger condition。

複合 trigger condition：以下の trigger 条件の複合を指定。

CHs OR：CH0 ～ 3 のどれかが満たされればトリガー。

CH2 & CH3：CH2 の条件と CH3 の条件が両方同時に満たされればトリガー。

CH0 & CH1：CH0 の条件と CH1 の条件が両方同時に満たされればトリガー。

All CH AND：全てのトリガー条件が満たされた場合のみトリガー。

Single/Continue：シングルトリガーと連続トリガーの切り替え。

Single = SW1 の Start ボタンを押した後、トリガー条件が満たされると

1 回のみ取り込み。(オシロのシングルモードに相当。)

Continue = SW1 の Start ボタンを押した後は、トリガー条件が満たされた時は

常に波形取り込み。(オシロの Normal モードに相当。)

トリガースキップ設定：後述。

SW5 は上記のパラメーター設定モードか、表示区間や表示倍率設定モードにするかの切り替えをする。もし、パラメーター設定モード(Power-ON default)なら、SW3,4 は上記の機能を行う。もし、表示区間設定モードなら、SW3,4 と Rotary Encoder は以下の機能になる。

表示区間設定状態

Rotary Encoder：表示区間、スタート点設定。

SW3：表示分解能設定 -> 高分解能

SW4：表示分解能設定 -> 粗

波形表示は以下のようにになっている。縮小率は Shrink と LCD に表示され、Shrink = 1/1 は表示画素がデータの 1 点に相当する。これが一番高い分解能。SW4 を押すごとに Shrink の値は増し、Shrink = 1/100 が最大となる。Shrink 値のデータ数が画素の 1 点として表示される。Shrink = 1/100 の場合は、表示された波形は 64k メモリー区間の全ての波形が表示されることになる。要は、64k 点の全波形を見たければ Shrink = 1/100 となる。Shrink が 1 以外では各画素点に集約されたデータ内部に H と L が混在している場合がある。その場合はピンク色の縦線が表示される。(1 か 0 かどちらもあり、という意味で。) その場合は SW3 を押し、分解能を上げていくと、1 と 0 の変化を視認できるようになる。

波形をどこから見るかは、ロータリーエンコーダーを回して、選択できる。なお、Shrink

= 1/100 の全波形表示の時はロータリーエンコーダーは機能しない。

データの取得

トリガー条件や波形解像度の設定後、SW1 を押すとデータの取得がスタートする。Single モードでは1回のみデータの取得を行う。その後、停止しているが、再度SW1 を押すと、1 回取得する。(オシロの Single モードに対応。) データの取得はトリガー条件を満たした場合のみ行われる。したがって、トリガー条件を満たすまで待ちになる。その際、永遠にトリガー条件を満たさない場合は、待ちに入ったままになるので、SW6 を押すと、強制終了するようになっている。(なお、SW6 で強制終了した場合はデータメモリーの内容は空のままなので、波形表示はされない。Continuous-mode で正常に終了するには SW2 を押す事。その場合、最

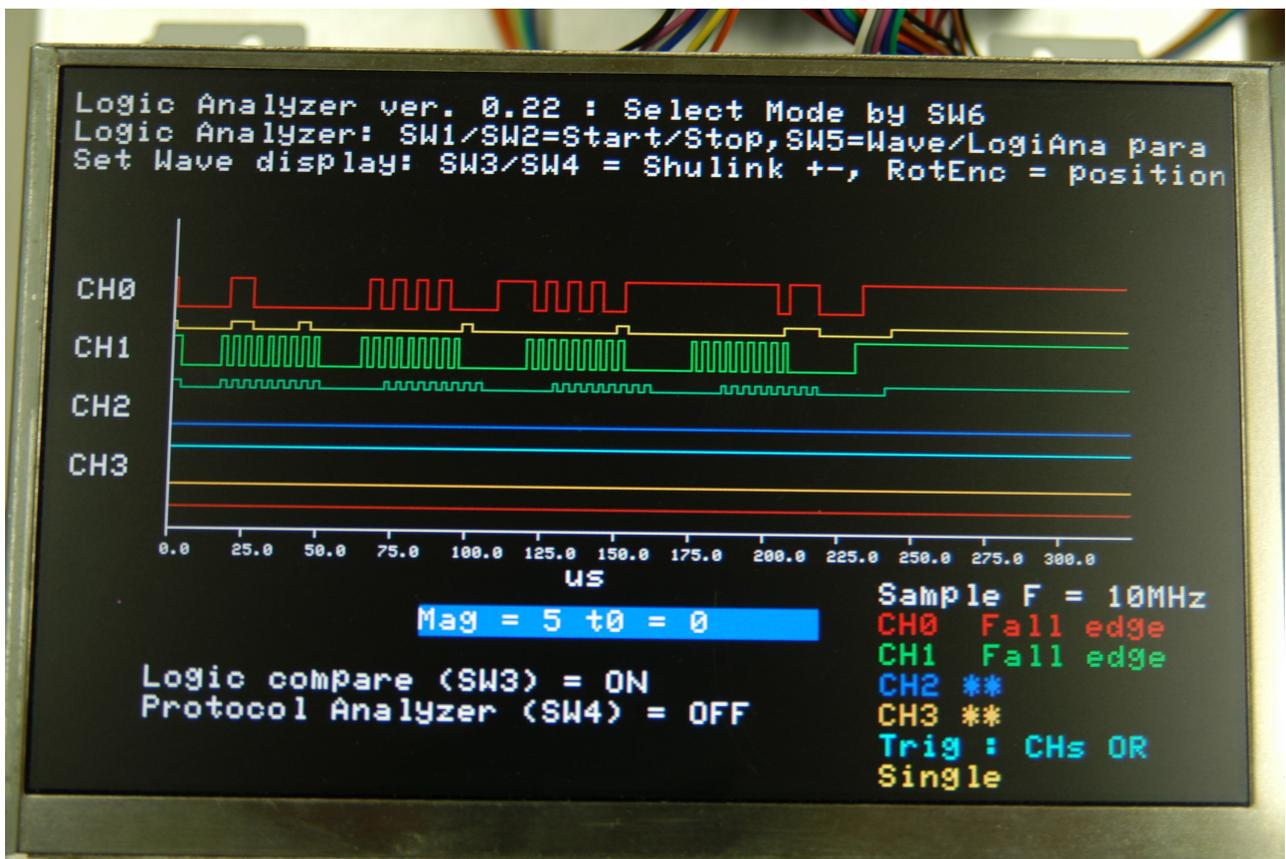


写真1 比較波形表示 (写真は ver.0.22 のもので ver.0.23 では少し異なる。)

後にトリガーがかかって、Continuous-mode が終了したデータが表示可能となる。)

Continuous モードではトリガー条件が満たされれば連続してデータを取得する。(オシロの Normal モードに対応。) これを止めるには SW2 を押すと、最後のデータ列を取得後、停止する。なお、SW6 で強制終了させることもできるが、その場合はデータメモリーの内容は保証されない。

トリガースキップ条件の設定 : (ver.0.4 より追加)

いろいろ使用していた結果、トリガー条件が満たされた場合でも、それをスキップして次の

トリガーからデータ取得をしたい場合が出てきたので、トリガースキップ・カウントの設定を追加した。SW3 や SW4 で設定項目を移動させていくと、Single/Continuous の次に、スキップ設定の項目に入る。スキップカウントは S xx という具合に表示される。ここで xx がスキップカウントであり、この回数、トリガー条件が満たされた後、初めてロジックデータの取得を始める。例えば、S 2 の設定では、2回、トリガーをスキップしてから波形取得をする。なお、S 0 はスキップを行わない。

このモードは例えば、以下のように使用すると便利である。マイコンの汎用入出力ポートでは、

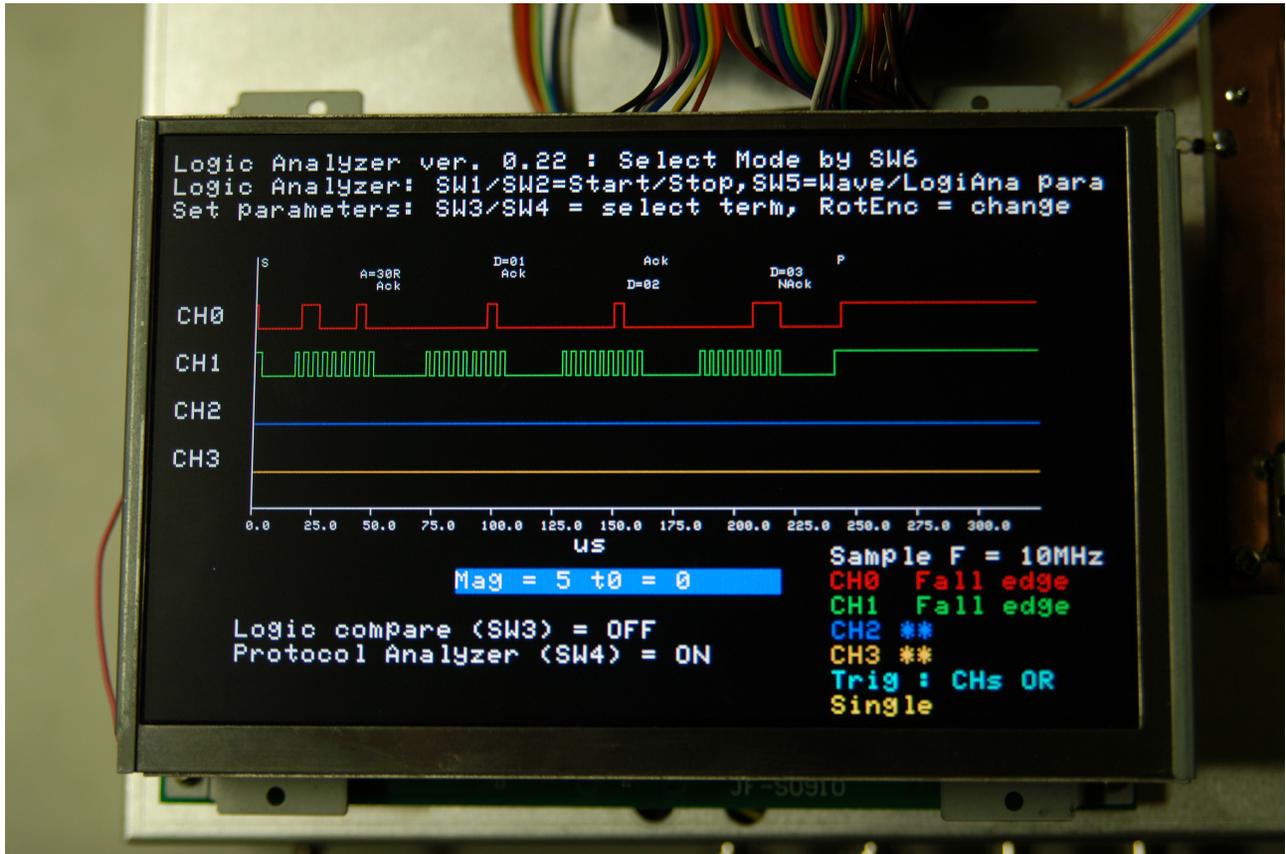


写真2 プロトコルアナライザー表示例。文字が小さいが赤い波形の上部に I2C プロトコルの S、Address+ R/W、Ack、Data 値等が表示されている。(写真は ver.0.22 のもので ver.0.23 では少し異なる。)

イニシャライズ時に一旦、L レベルになった後、正常に H 出力になったりする場合がある。この場合、そのポートで Fall edge でトリガーをかけるとパワーオン時やりセット時に、トリガーがかかってしまい、次に続く最初の有効な信号を取り込む事ができない。これを逃げるには、トリガーのスキップカウントを 1 にすればよい。

また、別の例では SPI 信号のモニターを行っている場合、20 番目の SPI 通信のやり取りが見たい場合がある。この場合は Single モードで /SS 信号でトリガーをかけ、スキップカウントを 20 とすれば詳細を見る事ができる。(なお、CPU の Power-ON 時にも /SS が L になる場合は、21 カウントを設定する。(注：スキップ機能は Continuous モードでも有効。))

スキップカウントは最大 2045 まで設定できる。スキップの判定は FPGA で行っており、サンプリング指定周波数の 3 クロックで判定されるため、高速である。

スキップカウント設定モードに入った場合、SW2 を押すと増分を 1, 10, 100, 1000 の範囲で変えることができる（ラウンドロビンで切り替わる）。増分は最終行の中央に白文字で表示される。その増分をもって、ロータリーエンコーダーで増減を行う。例えば、1001 カウントしたい場合には、まず、増分を 1 に選択し、ロータリーエンコーダーを右に回して 1 をセット、次に SW2 を 3 回押して増分を 1000 に設定、ロータリーエンコーダーを右に回して 1000 を加算。

再度、述べておくが、スキップを解除するにはスキップカウントを 0 とする。これにより、スキップせずに、トリガがあったと同時に波形が取得される。

2-4 波形比較、プロトコルアナライザーの設定

SW3, 4 で波形比較、プロトコルアナライザーの設定を ON/OFF する。（押すごとにトグル。）波形比較を ON すると、Mode0 で波形表示した時に、以下の写真のように各 CH の波形の下に高さが小さい比較波形が表示される。これにより、前に Mode1 で SD-Card にセーブしておいた波形と、現在の波形の違いを比較できるようになる。

比較波形のセーブは以下のようにする。Mode0 で波形を取得した後、SW6 で Mode1 に切り替え、SW1 を押す。これにより、現在の波形データが binary ファイルで SD-Card にセーブされる。ファイル名は Lwn.dat であり、n はファイル番号（0～9999）である。n は Mode1 下でロータリーエンコーダーで増減できる。（注：Mode2 下でのファイルセーブと異なり、セーブ後には n は自動インクリメントされない。）ファイルセーブにはやく 10 秒ほどかかるが、Mode2 での text ファイルセーブ（約 45 秒）よりは速い。なお、PC 側で SD-Card から binary ファイルを読み込んでグラフ表示する手段を確保すれば、この binary ファイルをグラフ表示に使用する事もできる。

過去にファイルをセーブしてあれば、そのファイル番号をロータリーエンコーダーで指定してやれば、昔のデータとの波形比較もできる。

プロトコルアナライザーを ON するには SW4 を押す（OFF / I2C / SPI の順で切り替わる）。プロトコルアナライザーが ON していると、Mode0 で以下の写真のように波形表示の上部にプロトコルが文字で表示される。現在の ver.0.4 のソフトでは I2C プロトコルと SPI のモード 0（SCK の立ち上がりでデータサンプリング）がサポートされている。なお、今後のソフトの変更により、他のプロトコル = SPI の各モードや RS232C などにも拡張できる。

注：I2C プロトコルの解析のためには、固定したチャンネル指定で CH0, CH1 入力に SDA, SCL を入力してやる必要がある。（ソフトがそのようにしてあるため。）

SPI プロトコルの解析のためには、固定したチャンネル指定で各チャンネルの入力に以下のように入力してやる必要がある。（ソフトがそのようにしてあるため。）CH0 = /SS, CH1 = SCK, CH2 = MOSI, CH3 = MISO。プロトコル表示では MOSI データは MO= の後に、MISO データは MI= の後に 16 進数 1 バイトで表示される。

注 2：サンプリング周波数が低かったり、波形表示の圧縮率が高い場合、正常に立ち上がりエッ

ジや立ち下がりエッジを認識できず、誤った解釈や認識不能のプロトコルが検出されることがある。その場合は、表示圧縮率を下げるか、サンプリング周波数を上げること。解釈不能なプロトコルは "X" と出力される。不正なプロトコルは "I" = illegal と表示される。従って、正しくプロトコルを表示するためには高いサンプリング周波数を使用すると良い。また、最初の方の不要なプロトコル部分を除くためにはトリガースキップ機能を使うのも有効である。

2-5 SD-Card へのデータ出力

モードを SW6 (SW5 を押したまま SW6 を押す) で切り替えて、SD-Card モードにすると SD-Card へデータを書き出すことができる。Windows の Text フォーマットでファイル出力される。ファイルのヘッダー部分には計測条件が書かれている。その後、4 チャンネル、64k 点のデータが続く。

ロータリーエンコーダーでファイル番号を変えることができる。なお、ファイル番号は 1 回ファイル出力するごとに自動的にインクリメントされる。この番号を戻したい場合などにロータリーエンコーダーを使う。

ファイル出力は SW1 を押すとスタートする。一端スタートするとファイル出力が終わるまでストップできない。

2-6 USB モード

SW6 (SW5 を押したまま SW6 を押す) で USB モードにすると、USB を経由して PC にデータを転送することもできる。HyperTerminal 等のソフトで受け取る。転送データは ASCII text である。HyperTerminal の設定では、「フロー制御なし」、とすること。ボーレートは現在のプログラムでは 9600 baud だが、プログラムを書き換えれば、より速い転送速度とできる。HyperTerminal で受信データをファイルへストアと設定しておく、ファイルにデータをセーブできる。

SW1 を押すと転送がスタートする。転送を中断したければ、SW2 を押す。

(3) 設計資料

どのようにしてロジアナを実現しているかについての概略を記しておく。(ここは ver.0.2 と同じ)

3-1 ロジアナ部

ロジアナ本体は DE0-nano ボード上の FPGA (Cyclone IVE) で実現されている。FPGA 内部の SRAM (今回は 4-CH なので、4-bit 幅) にデータは蓄積される。データ取得が開始すると、指定したサンプリング周波数で常に SRAM にデータを書き込んでいる。トリガー条件が満たされたなら、そこから 64k - 3 点を書き込んだ後、計測を終了し、AVR 側に計測の終了を通知する。- 3 点、データが少ないのは、トリガー条件が満たされる 3 点前までのデータもモニターできるようにするためである。トリガー条件が満たされたかどうかの判定は FPGA 上

にインプリメントされたロジックで判定する。

以上より、データの蓄積されている高速 SRAM は FPGA 内部にあり、AVR 側はその SRAM のどこのアドレスからデータを読むかを指定して、波形表示の処理を行う。FPGA 側ではトリガーがかかった時のメモリーアドレスを記録しており、まず、そのアドレスを AVR 側が読み、それに表示区間の指定オフセット値を+したアドレスからデータを読み出す。FPGA 内部のメモリーは 64k であり、64k アドレス空間をラウンドロビンしてデータの蓄積をしているので、それを考慮して読み出す。

3-2 カラーグラフィック液晶制御部

カラーグラフィック液晶の制御も DE0-nano ボード上のロジアナ部と同じ FPGA が行っている。FPGA は並列処理ができるので、まったく異なったモジュール（液晶制御とロジアナ）をインプリメントして同時並列に動作させることができる。カラーグラフィック液晶は画像メモリーが必要であるが、DE0-nano ボード上に搭載された SDRAM を利用すれば、いとも簡単に画像メモリーを実現できる。SDRAM の制御も FPGA が行っている。SDRAM には refresh 等の休止期間が必要であるので、FPGA 内部の SRAM の一部を double buffer とし、そのうちの片側の SRAM がカラーグラフィック液晶に画素データを掃き出している時間中にもう片方の SRAM に SDRAM からバースト転送を行い、かつ、余った時間で refresh 処理を行う。これをカラーグラフィック液晶の水平区間で切り替えて繰り返す。カラービットは 16-bit 幅であり、SDRAM, bouble-buffer SRAM のデータ幅も 16-bit となっている。

3-3 AVR ボード

ピン数が多く、内部 RAM 容量も多い、ATMega1284 を使った。SD-Card や USB I/F (FT232R) の制御の他、ロジアナのユーザーインターフェースであるスイッチ類の読み取り、および液晶へのグラフ表示のためのデータ処理等を行う。SD-Card, FPGA との通信は主に SPI I/F で行う。なお、FPGA 内部の SRAM にストアされたデータの読み出しのみ、スピードを重視してパラレル I/F とした。

プログラムは BASCOM-AVR により記述。この方が短時間で作りやすい。SD-Card の Windows ファイル処理のために、AVR-DOS を使用（BASCOM で使用できる。）USB 通信は、BASCOM で単に Print 文を使えばよい。

3-4 拡張性

AVR のプログラムを書き換えれば、I2C や SPI 以外のプロトコルアナライザーも可能。Mega1284 における現在の flash 占有率は 36% なので、プログラム容量は大幅に余っている。

DE0-nano の FPGA のゲート占有率等は logic elements = 17%, Total memory bits = 51%.

データチャンネル数は FPGA の SRAM が足りさえすれば拡張可能。なお、現在の要求はシリアルライン（I2C や SPI I/F）などのモニターのため、4-CH で十分足りている。

データメモリーの容量は DE0-nano ボードの SDRAM（32MB）をうまく使えば 16M 程度

まで拡張できる可能性有り。ただし、このSDRAMはグラフィック液晶表示に使う frame buffer(1枚で 1MB) にも使っているため書き込みタイミングに制限が出て、ロジアナの波形メモリーも兼用すると、ロジアナの sampling 周波数はあまり上げられない可能性が出る。100MHz 程度までか？(液晶表示のダブルバッファ容量を増加し、ロジアナ部からの書き出しに FIFO を設置。)でも 16MB のメモリー長は魅力。今後、必要性が出たら設計するが。ただし、50MHz なら、AVR や PIC では十分と思われる。なお、現在は FPGA 内部メモリーを使う 64k x 4CH メモリーでも実用上十分なため、特には拡張は考えていない(設計は結構面倒かも?)。これに対し、ver.0.4 でのトリガースキップ機能の追加により、あまり長いデータバッファは不要になってきたし、その機能により長いデータメモリー中をスクロールして探すという面倒な作業を省けるようになった。(再度述べるが、スキップ機能は FPGA 上に実装しているので最低 3 サイクルのサンプリング周波数で機能するので、AVR の処理を必要とせず高速であり、サンプリング周波数を十分に高速にしておけばスキップ中のトリガー見逃しが無い(例えば 50MHz sampling CK 設定なら、3CK=60ns。))

AVR の代わりに FPGA にその上の組み込み CPU である NIOS を搭載するか？ そのためには以下の検討が必要。AVR ボードではロータリーエンコーダー、6つのスイッチ、SD-Card I/F、USB I/F などの制御を行っており、それらを NIOS に任せると、いろいろな処理ルーチンを新たに作成する必要がある。また、BASCOM での波形表示処理ルーチンの一部では浮動小数点数の処理なども行っており、有料の NIOS ライセンスが必要となる。さらに、AVR 内部の SRAM も 6kB 程度使用しているため、AVR を NIOS に変えた場合、NIOS がその他に必要な SRAM と併せて、FPGA 内部の SRAM をロジアナ以外に NIOS 向けに十分割り当てられるかどうか分からない。さらに RAM を SDRAM に割り当てるとグラフィック液晶のフレームバッファとの調停が必要となる。いろいろ設計を試してみれば DE0-nano のみで液晶表示器付のロジアナができる可能性はあるが、けっこう面倒な考察が必要となり、短期間で簡単に試作するという趣旨からははずれるので、試していない。なお、DE0-nano より上位の FPGA ボードを使えば可能かもしれないが価格は格段に上昇する。

USB I/F の転送速度は FT232H を使えば、高速化が可能。そのための信号線を AVR ボードには出してある。ATMega の USART による SPI モードを使用できるように、XCK = SCK, /SS 用に 1 ポート (Test LED 兼用) を出してある。

AVR でやっているのだから CPU クロックが低く、ソフトの処理速度は、そこそこである。それを高速にしたい場合には、FPGA 内部に NIOS をインプリメントするか、外部の CPU (AVR) を CPU クロックがより速い ARM (例えば Cortex M4) にするのが良いかもしれない。(なお、安いタイプの ARM のみで高速ロジアナを作成するには少し無理がある。そうした場合には、おそらく 10MHz sampling の正確な間隔の波形取得や 200MHz sampling は不可能なので、FPGA は使わざるを得ないと思われる。)

以上。