

令和 2 年度卒業論文

シリコンラインセンサ読出高速化にむけた
FPGA による制御システムの開発

広島大学理学部物理科学科 4 年
クォーク物理学研究室
学籍番号 B174160

渡部 裕

2021 年 2 月 25 日
指導教官 山口 頼人 准教授
主査 山口 頼人 准教授
副査 水野 恒史 准教授

概要

FPGA を用いてシリコンラインセンサのシリアル読み出しの高速化に向けたセンサ制御システムの開発を行った。FPGA は Xilinx 社の Artix-7 を、その評価ボードは Digilent 社の NEXYS A7 を用いた。FPGA とは論理回路をプログラムすることで様々な機能を実装することができる集積回路である。その利便性及び優れた高速性から物理実験において検出器の読み出しに広く使用されている。シリコン検出器は γ 線や荷電粒子の代表的な検出器であり、扱い易いことから医療・産業界の X 線非破壊検査器へと応用されている。本研究では密閉容器の X 線検査装置として実用されている 256 ch シリコンラインセンサの制御用信号生成部とセンサ出力信号処理およびデータ転送の集約化を行った。これらの回路は従来独立なものであったが、FPGA によるデジタル回路により実現することができた。1 MHz のクロック信号を元に生成した種々の制御信号でセンサを駆動し、センサからの出力としてのアナログ信号を AD 変換した。その後データを ASCII コードに変換し PC に取り込むための UART 通信を実装した。現在、このシステムの読み出し速度の上限は UART 通信が決めており、より高速かつ大容量通信を可能とする Ethernet 通信の検討をしている。

目次

| | | |
|-------|------------------------|----|
| 1 | 序論 | 6 |
| 1.1 | シリコン検出器 | 6 |
| 1.1.1 | シリコン検出器の原理 | 6 |
| 1.2 | 現行の X 線検査システム | 7 |
| 1.3 | 256 ch シリコンラインセンサ | 8 |
| 1.4 | FPGA | 9 |
| 1.5 | AD コンバータ | 9 |
| 1.6 | 本研究の目的 | 10 |
| 2 | 開発 | 10 |
| 2.1 | 開発環境 | 10 |
| 2.1.1 | XADC | 11 |
| 2.2 | 開発の流れ | 11 |
| 2.2.1 | HDL を用いた回路記述・機能の設計 | 12 |
| 2.2.2 | シミュレーションによる動作確認 | 12 |
| 2.2.3 | 論理合成 | 12 |
| 2.2.4 | 配置配線 | 12 |
| 2.2.5 | FPGA への実装 | 12 |
| 2.3 | 制御信号の生成 | 13 |
| 2.3.1 | クロック信号 | 13 |
| 2.3.2 | リセット信号 | 13 |
| 2.3.3 | ゲイン信号 | 13 |
| 2.3.4 | バッファの使用 | 13 |
| 2.4 | データ読み出しシステムの開発 | 16 |
| 2.4.1 | UART | 16 |
| 2.4.2 | ASCII コード | 16 |
| 2.4.3 | データ通信 | 16 |
| 2.5 | XADC の使用 | 20 |
| 2.5.1 | XADC の較正 | 22 |
| 2.6 | センサ読み出しシステム | 24 |
| 2.6.1 | XADC とセンサの VIDEO 信号の同期 | 24 |
| 2.6.2 | センサのチャンネル情報 | 27 |
| 2.6.3 | FIFO へのデータの書き込みについて | 27 |
| 2.6.4 | UART でのデータ読出の速度について | 27 |
| 3 | 結果 | 27 |
| 3.1 | 暗電流 | 27 |

| | | |
|-----|----------------------|----|
| 3.2 | センサによる光の検出 | 30 |
| 4 | 今後の課題 | 34 |
| 5 | まとめ | 34 |

図目次

| | | |
|----|---|----|
| 1 | pn 接合フォトダイオード | 7 |
| 2 | p 型半導体と ngata 半導体のフェルミ準位 | 7 |
| 3 | 現行システムの構成の概要 | 8 |
| 4 | センサのタイミングチャート | 8 |
| 5 | アナログ値と変換後のデジタル値の関係 | 10 |
| 6 | FPGA ボード：Nexys A7 | 11 |
| 7 | 3 bit カウンターのシミュレーション結果 | 12 |
| 8 | 1 MHz の生成 | 13 |
| 9 | シングルエンドと差動伝送 | 14 |
| 10 | 終端抵抗と駆動電流 | 14 |
| 11 | FPGA から出力された制御信号（センサ未接続） | 15 |
| 12 | FPGA から出力された制御信号（センサ接続時） | 15 |
| 13 | データ通信の流れ | 17 |
| 14 | エンコーダーのタイミングチャート | 17 |
| 15 | エンコーダーの信号の説明 | 18 |
| 16 | 送信期のタイミングチャート | 19 |
| 17 | 送信期の信号の説明 | 19 |
| 18 | XADC のポート [3] | 20 |
| 19 | DRP とレジスタ [4] | 21 |
| 20 | DRP のタイミングチャート [5] | 22 |
| 21 | 400mV の電圧をかけた時の XADC の値 | 23 |
| 22 | XADC の標準偏差 | 23 |
| 23 | 電圧と XADC の値の関係 | 24 |
| 24 | イベントサンプリングモードのタイミング [7] | 25 |
| 25 | convst 信号のタイミング | 26 |
| 26 | FPGA とシリコンラインセンサのセットアップ | 27 |
| 27 | 暗箱を閉めた様子 | 28 |
| 28 | FPGA 部分 | 28 |
| 29 | 暗電流における 256 ch のヒストグラム | 29 |
| 30 | 暗電流状態でのセンサからの VIDEO 信号（5120 セットの平均） | 29 |
| 31 | 暗電流状態でのセンサからの VIDEO 信号のプロット | 30 |
| 32 | 暗電流状態でのセンサからの VIDEO 信号のプロット（拡大） | 30 |

| | | |
|----|---|----|
| 33 | 暗電流におけるチャンネルごとの標準偏差 | 30 |
| 34 | ラインセンサ | 31 |
| 35 | ねじ穴から光を入れた時のセンサの応答 (5120 セットの平均) | 31 |
| 36 | 光を入れた状態での 5120 回分のプロット | 32 |
| 37 | ねじ穴から光を入れた時のチャンネルごとの標準偏差 | 32 |
| 38 | 光を当てた時の暗電流によるバックグラウンドを引いた時のヒストグラム | 33 |
| 39 | 蓄積時間の違いによる VIDEO 信号の変化 | 33 |
| 40 | 蓄積時間とヒストグラムの面積 | 34 |

1 序論

1.1 シリコン検出器

シリコン検出器は安価でエネルギー分解能、位置分解能が高く（電子-正孔対を生成するのに必要なエネルギーは 3.62 eV[1] であり、ガスではイオン化エネルギーは 26 eV 程度 [2]）加工が容易（高純度単結晶の生成が可能）なため、物理実験において X 線や γ 線、高エネルギー粒子の検出に広く使用されている。ALICE 実験ではシリコンピクセル検出器が用いられており、荷電粒子の衝突や飛跡の測定には不可欠な存在である。シリコン検出器は物理実験のみならず、医療における PET・CT 検査用の X 線検出器や、文化財・建築物の非破壊検査などにも使用されている。加えて、デジタルカメラなどに用いられる CCD イメージセンサや CMOS イメージセンサもシリコン検出器の一種であり、この様にシリコン検出器は幅広い分野使用されている。

1.1.1 シリコン検出器の原理

シリコンなどの半導体は伝導帯に電子が存在しない為、電圧を印加しても電流は流れない。しかし、絶縁体に比べ禁制帯が狭く、光電効果や放射線との相互作用などで電子が伝導帯へ遷移することができる。しかし、真性半導体では伝導帯に励起する電子の数は少ない。そこで、半導体よりも価電子が多いドナー（リンなど）や、価電子が少ないアクセプター（ホウ素など）と呼ばれる不純物を添加することで電気伝導率を上昇させる。ドナーを添加したものを n 型半導体、アクセプターを添加したものを p 型半導体という。この時のフェルミ準位は図 2 の様になり、n 型半導体ではドナーのフェルミ準位は伝導帯に近い場所にあり、p 型半導体ではアクセプターは価電子帯に近い場所にできる。従って、n 型半導体では電子が伝導帯に遷移しやすくなり電気伝導率が上昇する。ここでのフェルミ準位とはフェルミディラック統計における電子の存在確立が 50 % になるエネルギー準位のことである。p 型半導体ではアクセプターのフェルミ準位に価電子帯の電子が遷移することで価電子帯に正孔が生じ、電気伝導率が上昇する。この時発生した電子や正孔を検出することで、シリコン検出器は X 線などを計測する。フォトダイオードなどの光検出器として使用する場合、pn 接合フォトダイオード（図 1）や pin 型フォトダイオードなどが存在する。pn 接合フォトダイオードでは、p 型と n 型の接合面で電子と正孔が結合し、キャリアの存在しない空乏層ができる。空乏層の n 型との境界付近ではドナーから電子が不足しドナーイオンに、p 型との境界付近ではアクセプターから正孔が不足しアクセプターイオンになり内蔵電場が発生する。空乏層に光が入射すると、光電効果により空乏層中の電子が励起し、キャリアとしての電子と正孔が生成される。生成されたキャリアは内蔵電場により電極に集められ電流が流れる。この電流により X 線を検出する。pin 型フォトダイオードは pn 接合の間に絶縁性の i 型半導体（真性半導体）を挟んだもので、i 型半導体が空乏層の役割をはたす。pin 型では内蔵電場が発生しない為、外部から電圧をかけることで光電効果により発生したキャリアを加速する。

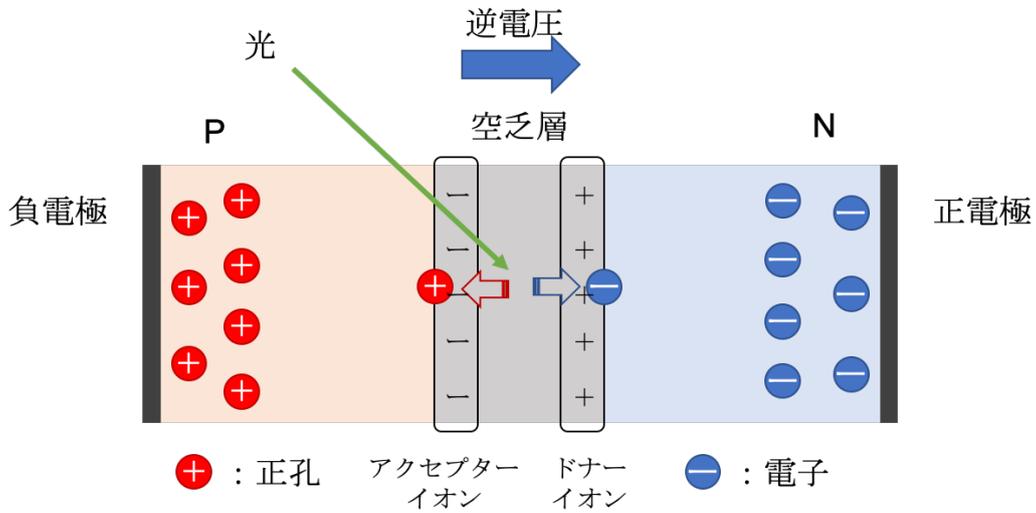


図1 pn 接合フォトダイオード

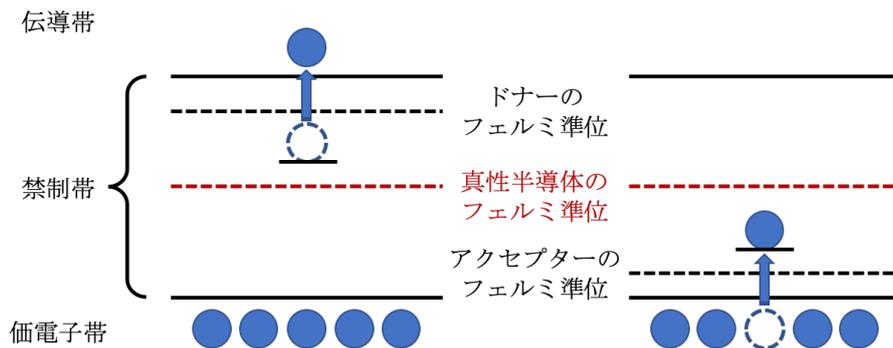


図2 p型半導体と ngata 半導体のフェルミ準位

1.2 現行の X 線検査システム

本研究で使用するサイエナジー社が製造する X 線センサシステム LINE 型の構成の概要を図 3 に示す。現行の X 線システムは大きく分けて、センサヘッド、センサコントローラ、X 線源の 3 つから成る。センサヘッドは 256 ch のフォトダイオードを持つシリコンラインセンサとそれを制御するための駆動回路からなり、センサコントローラからの制御信号のタイミングで VIDEO 信号（フォトダイオードに溜まった電荷を電圧 (0-5V) に変換した信号）をシリアルに出力する。CPLD (Complex Prgramable Logic Cevice) は FPGA と類似した機能を持つが回路規模は FPGA より小さく、制御信号の生成などに使用される。センサコントローラは CPU、AD コンバータなどから構成され、X 線発生装置の制御及びセンサ制御信号の生成、VIDEO 信号の AD 変換、画像処理、外部機器との通信などを行う。現行システムでは制御信号のクロックは 1 MHz で生成されている。本研究ではデジタル回路を FPGA に集約することを目指す。これは今後の FPGA によるセンサ制御システムの高高速化や、FPGA による画像処理を行うための基礎研究である。

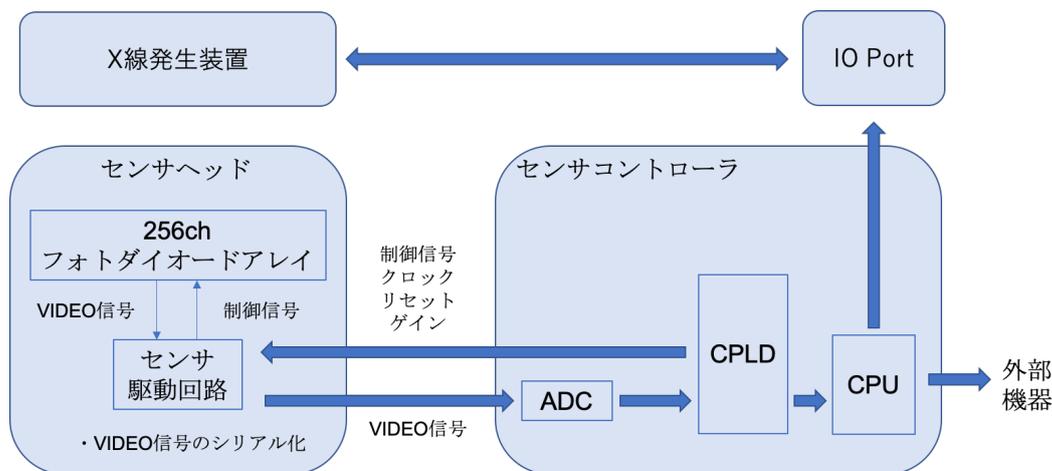


図3 現行システムの構成の概要

1.3 256 ch シリコンラインセンサ

今回使用するサイエナジー社の X 線センサシステム LINE 型は、センサヘッドとして 256 個のフォトダイオードが一列に並んだシリコンラインセンサを採用している。センサからのデータはクロックに同期され、シリアルに出力される。センサは 1 MHz のクロック信号 (CLK) と、それに同期したリセット信号 (RST)、及びゲイン信号 (GAIN) の 3 つの信号で動作する。クロック信号はシリコンラインセンサが動作する上でタイミングの基準となる信号であり、リセット信号はシリコンラインセンサの動作開始の合図とフォトダイオードの蓄積時間を制御するための信号である。ゲイン信号はシリコンラインセンサから出力される VIDEO 信号の振幅を 2 段階で調整し、ゲイン信号が High のとき VIDEO 信号の振幅はゲイン信号が Low の時の 2 倍になる。制御信号のタイミングチャートを図 4 に示す。センサーはリセット信号が High から Low に変化すると、次のクロックの立ち下がりで動作を開始する。リセット信号が Low になってから 20 クロック目と 21 クロック目に最初の VIDEO 信号が出力され、以降 4 クロックのサイクルで VIDEO 信号が出力される。1040 ~ 1041 クロックで最後の VIDEO 信号が出力されると、センサは次のリセットの立ち下がりまで待機する。リセット信号が High の間、フォトダイオードは信号を蓄積する。リセット信号が High に変化してから 8 クロック目に蓄積を開始し、リセット信号が Low に変化してから 8 クロック目で蓄積を終了する。

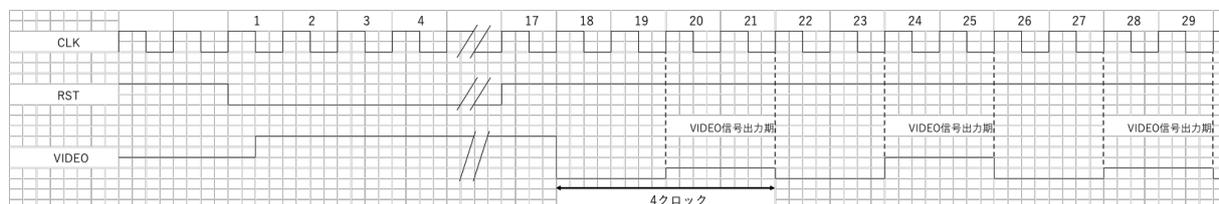


図4 センサのタイミングチャート

1.4 FPGA

FPGA(Field Programmable Gate Array)とは論理回路をプログラムすることで、様々な機能を実装できる集積回路の一種である。FPGAはあらかじめ用意されている基本的な回路を組み合わせてことができ、何度でも変更できる。一方、CPUは回路構成は最初から決まっておき、変更することはできないがメモリに保存された実行プログラムにより様々な機能を実装する。CPUの場合でもソフトウェアのプログラム次第で様々な機能を実装できるが、ハードウェアの回路の特性などで作業によっては効率が悪くなったり、使われない回路が増えるなど非効率なものとなってしまう。ASICの場合、目的にあった専用の回路のみを実装するので、無駄がなくFPGAよりも高速で動作するが、量産しない場合などにはコストが高くなる傾向があり、一度作成すると後から変更することができない。FPGAはCPUに比べ、ハードウェアを効率化させることで消費電力や発熱を抑えることができ、並列処理により高速性にも優れている。FPGAは物理実験において、高速ADCのインターフェースや加速器の制御などに用いられている。FPGAやCPUなどはクロックと呼ばれるパルス信号に同期して演算を行う。CPUのクロックは高速なもので ~ 5 GHzであるのに対し、FPGAのクロックは数100 MHzと一桁近く遅い。しかし、FPGAは演算機がCPUに比べ格段に多く、並列処理を行うことでCPU以上の高速性を可能にしている。このことから、FPGAは並列度の高い処理を行うことに適している。並列性の高い処理としては画像処理やパターンマッチングなどが挙げられる。

1.5 ADコンバータ

電圧、電流などのアナログ情報をデジタル情報に変換するのがADコンバータである。AD変換はAD変換の対象をある時間ごとに標本化し、比較器を用いて量子化することで行う。標本化とは入力信号がAD変換の途中で変化しない様に一時的に保持する為の作業であり、比較器とは入力電圧を基準電圧と比較し、基準電圧より高いか低いかを判定する回路のことである。アナログ値とデジタル値の関係を図5に示す。図5は入力電圧(0~1 V)を3ビット分解能でAD変換したと仮定した場合のグラフである。この時、変換されたデジタル値の上位1ビットをMSB(Most Significant Bit)、下位1ビットをLSB(Least Significant Bit)と呼ぶ。LSBはADコンバータの最小分解能の幅と等しく、図5の場合LSBは ~ 0.143 Vである。ADコンバータにはフラッシュ型、パイプライン型、逐次比較型、 $\Delta\Sigma$ 型などがある。今回は基本的なフラッシュ型と逐次比較型について説明する。フラッシュ型はNビット分解の場合、 $2^N - 1$ 個に分圧された基準電圧と比較器を用いて、一度にAD変換を行う。フラッシュ型は最も高速(数GHz)で動作するが、分解能が上がると必要な基準電圧、比較器の数も急激に増加するため、分解能は8ビットが限界である。逐次比較型は標本化したアナログ入力をMSB側から順に1ビットずつ変換し、分解能は10~18ビットでフラッシュ型よりも高分解能である。Nビット分解能の場合、AD変換に最低でもN回のクロックサイクルが必要な為、変換速度はフラッシュ型に比べると遅くなる(数100 kSPS~10 MSPS)。

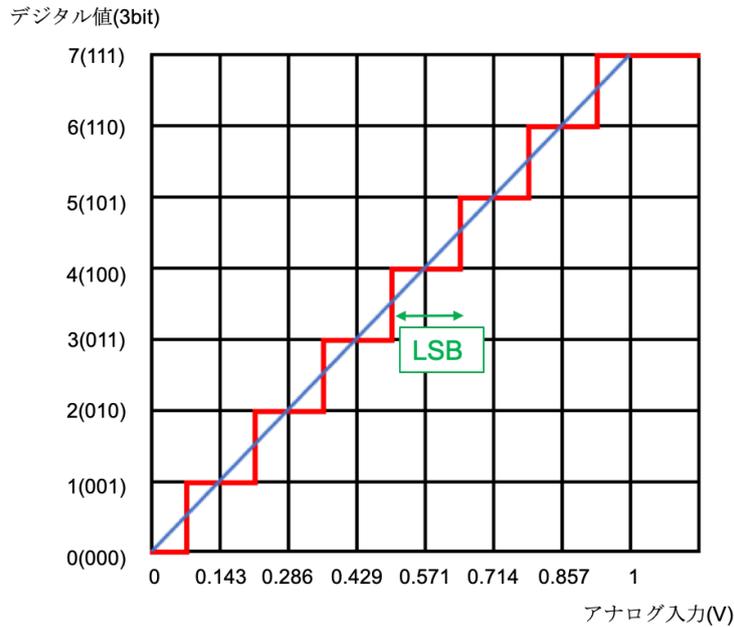


図5 アナログ値と変換後のデジタル値の関係

1.6 本研究の目的

本研究の目的は、FPGA を用いて産業用 X 線検査システムの制御を行うことである。FPGA を用いることで、データ処理の高速化に向けたデジタル回路の集約を行う。FPGA の柔軟性を活かすことで、これまで複数のチップで構成されていた回路を1つの FPGA チップに実装することを目指す。

2 開発

2.1 開発環境

本研究では FPGA に Xilinx 社の Artix-7 とその評価ボードである Digilent 社の Nexys A7(図 6) を使用した。FPGA の回路設計および実装には Xilinx 社の Vivado HLx edition Webpack を使用した。センサからの VIDEO 信号の AD 変換には、Artix-7 のオンチップ ADC である XADC を用いた。

2.2.1 HDL を用いた回路記述・機能の設計

Vivado ではテキストファイルにより回路情報を記述する。記述には Verilog HDL と呼ばれるハードウェア記述回路を用いる。Verilog HDL では回路ブロック（回路を機能ごとに分割したもの）を module という単位で表し、作成した module を複数組み合わせることで目的の機能を実現する。Vivado にはすでに設計された機能、IP(Intellectual Property) が用意されており、IP と自作した module を使いながら設計を行う。

2.2.2 シミュレーションによる動作確認

HDL で設計した回路の動作をシミュレーションによる確認する。シミュレーションでは入力信号を生成し、それに対する出力信号の結果を得る。図 7 は 3 ビットカウンタのシミュレーションの例である。ソースコードはリセットが Low の場合、カウンタを 0 にリセットし、High の場合、クロックの立ち上がりでカウンタを +1 する。図 7 では clk_100 と rst がシミュレーションで生成された信号で、counter が出力である。

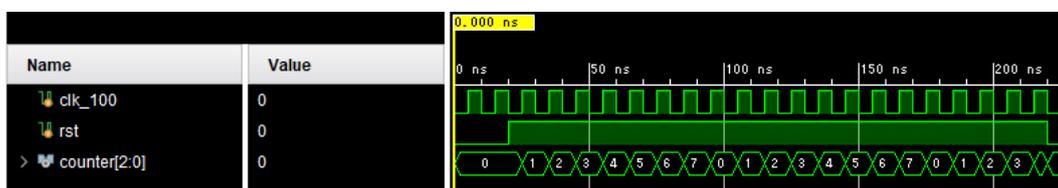


図 7 3 bit カウンタのシミュレーション結果

2.2.3 論理合成

機能の設計で説明した Verilog HDL はレジスタ転送レベルと呼ばれ、抽象的な記述を行う。ハードウェア記述言語には抽象度の高い順から動作レベル、レジスタ転送レベル、ゲートレベルがある。動作レベルではハードウェアの動作やアルゴリズムをそのまま記述したもので、フローチャートなどをそのまま記述する。レジスタ転送レベルではレジスタを定義し、レジスタとレジスタ間の接続を記述する。ゲートレベルでは AND ゲートや OR ゲート、NOT ゲートのみで構成させる記述である。論理合成では、レジスタ転送レベルの抽象的な回路動作の記述から論理回路を生成する。

2.2.4 配置配線

配置配線とは、論理合成で生成された論理回路を実際に使用する FPGA にどのように実装するかを決める作業である。FPGA は基本的な回路と回路を接続する為の配線があらかじめ用意されており、それらの回路を組み合わせることで目的の回路を実装する。配置配線で、論理合成により生成された回路を FPGA 内のどこに配置し、回路同士をどのように配線するか、また電圧レベルなどを決定する。

2.2.5 FPGA への実装

配置配線まで終了すると必要な回路情報は全て完成する。完成した回路情報を FPGA に実装するためには、bit ファイルを生成し、FPGA にダウンロードする。Nexys A7 では、JTAG ポートを経由して bit ファイルをダウンロードする。

2.3 制御信号の生成

2.3.1 クロック信号

制御信号は FPGA により生成するが、使用する FPGA ボードの Nexys A7 は 100 MHz のクロックを出力する。一方、センサーは 1 MHz のクロックに同期して動作する。従って、100 MHz のクロックを 1/100 分周することでセンサ用の 1 MHz クロックを生成する。7 ビットのレジスタ（カウンター：2 段目）と 1 ビットのレジスタ（CLK1MHz：3 段目）を用意する（図 8）。カウンターはクロックの立ち上がりに同期して常に +1 されており、クロックの立ち上がりでカウンターの値が 99 の時に 0 にリセットされる。レジスタ（CLK1MHz）はクロックの立ち上がりでカウンターの値が 49 の時に Low に遷移し、カウンターの値が 99 の時に High に遷移する。この流れを繰り返すことで、レジスタ（CLK1MHz）は 1 MHz のクロックとして機能する。

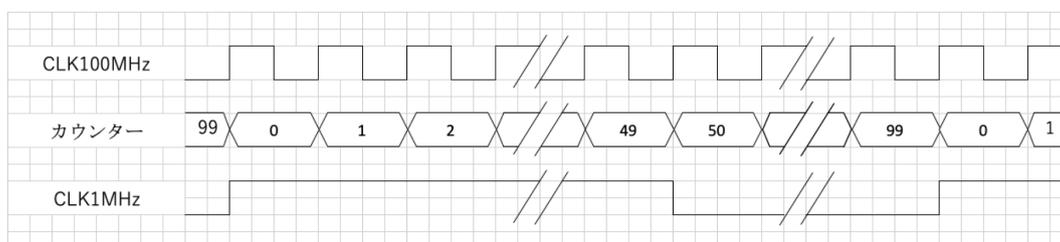


図 8 1 MHz の生成

2.3.2 リセット信号

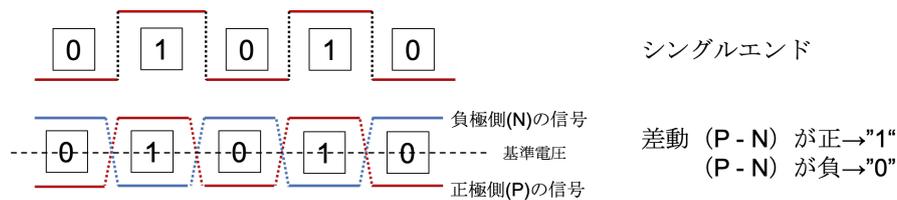
リセット信号用に 1 MHz のクロックをカウントする。カウントはクロックの立ち上がりに同期しており、256 ch 分の VIDEO 信号が出力された後 (>1041 カウント以降) にリセットされる。カウントが $n (> 1041)$ の時にリセットされる場合を考える。リセット信号は始めは High の状態でスタートし、カウンターの値が n の時に High から Low に変化する。その後、カウントが 15 の時 Low から High に変化する。

2.3.3 ゲイン信号

ゲイン信号は非同期の制御信号で、センサからの VIDEO 信号の振幅を 2 段階で制御する。ゲイン信号が High の時、VIDEO 信号の振幅はゲイン信号が Low の時の振幅の 2 倍になる。本研究では、GAIN は常に High の状態（VIDEO 信号の振幅 2 倍）で使用。

2.3.4 バッファの使用

デジタル回路における信号の伝達にはシングルエンド方式と差動伝送方式がある。シングルエンドとは 1 本の信号線を使用し、グラウンドに対する信号線の電圧差で High と Low を識別する。差動伝送では 1 つの信号につき正極と負極で 2 本の信号線を使用し、信号線の電圧差で High と Low を判別する。従って、差動伝送は外部からのノイズに強く、高速なデータ通信などに用いられる。（図 9）



差動では外部からノイズが混入しても、二つの信号線の電圧差は変化しない

図9 シングルエンドと差動伝送

今回使用するシリコンラインセンサの制御信号（CLK、RST、GEIN）は差動信号のため、先述した制御信号を正極の信号として、正極の信号を反転させたものを負極側の信号として生成する。信号の電圧レベルは Nexys A7、センサの両方で 3.3 V レベルである。差動伝送は高速なデータ通信で用いられることが多いため、インピーダンスの違いによる反射などを防ぐために 100 Ω の抵抗で終端されている。信号の電圧レベルが 3.3 V の場合、オームの法則より電流は 33 mA 流れる必要がある（図 10）。Nexys A7 では I/O ポートとして Pmod が用意されており、制御信号は Pmod から出力する。出力した制御信号（CLK、正極側）をオシロスコープで観測した時の様子を図 11 と図 12 に載せる。図 11 はセンサに接続していない状態のクロック信号を観測したもので、クロック信号は 3.3 V と 0 V 電圧を示している（正常）。一方図 12 はセンサと接続した状態でクロック信号を観測したもので High の時で 2.05 V、Low の時で 1.45 V を示しており、正常に動作していない。これより、Nexys A7 の駆動電流は 33 mA よりも低く、Nexys A7 単体ではセンサを正しく動作させることができないことが判明した。

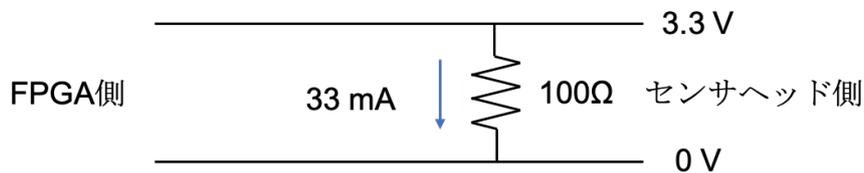


図 10 終端抵抗と駆動電流

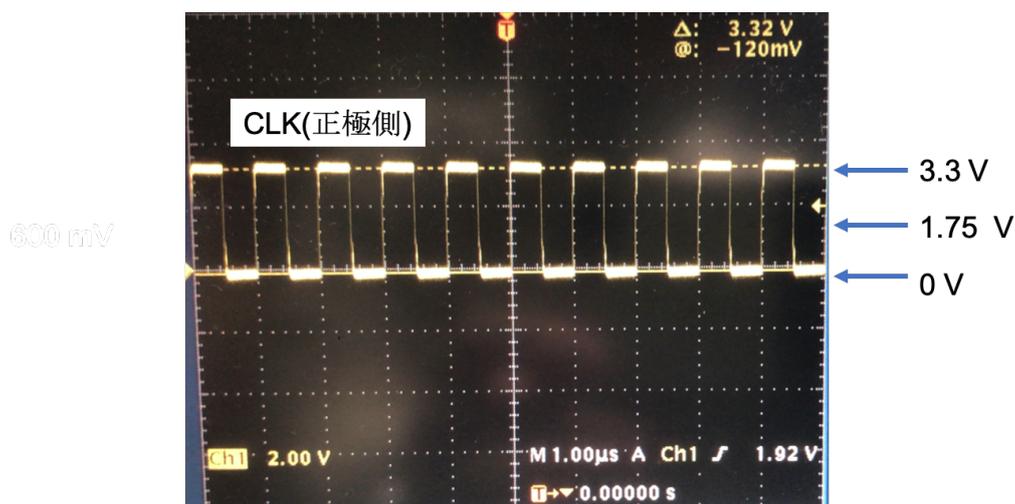


図 11 FPGA から出力された制御信号（センサ未接続）

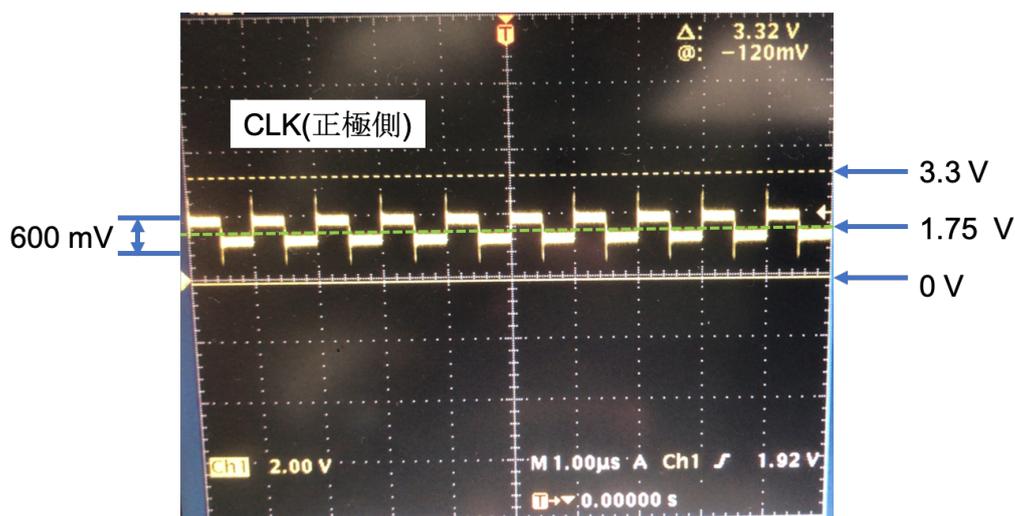


図 12 FPGA から出力された制御信号（センサ接続時）

制御信号を正しく出力するためにバッファを使用する。ここでのバッファとは入力された電圧をそのまま出力する回路のことである。本研究では、バッファとして Texas Instruments 社のラインドライバである AM26LV31IDR を使用した。AM26LV31IDR はシングルエンドのデジタル入力を差動信号に変化するチップであり、駆動電流が 30 mA と強力である。AM26LV31IDR を使用することで、センサを接続した事による制御信号の電圧変化を防ぐことが可能である。

また、同じ理由で、シリコンラインセンサーからの VIDEO 信号もバッファを用いて電流を増幅する必要がある。こちらのバッファには ANALOG DEVICES 社の差動入力、差動出力オペアンプである LTC6362 を使用した。

2.4 データ読み出しシステムの開発

XADC で取得したデータを FPGA から PC に取り込むために、UART による ASCII コードでのデータ読み出しシステムを作成した。XADC からのデータを 10 進法に変換し、ASCII コードに変換して UART で送信する。受信側には、Linux PC (CentOS7) を使用し、端末上で cu コマンドを使用してデータを受信する。

2.4.1 UART

UART (Universal Asynchronous Receiver/Transmitter) とは非同期のシリアル通信の一種であり、送信用と受信用の二本の信号線を用いて通信する。非同期のためクロック信号が存在せず、通信する場合は送信側と受信側で通信速度 (ボーレート) を決める必要がある。このボーレートにより 1 ビット分の時間が決まる。UART では通信しない間、信号線は High に固定されており、通信する場合は、信号線を 1 ビット分 Low にする (スタートビット) ことで通信を開始する。スタートビットの後にデータが送られる。データ幅は基本的に 8 ビットである。データが送られた後、信号線が 1 ビット分だけ High になり (ストップビット)、信号線はそのまま High の状態を継続する。これにより、次の通信を開始することができる。

2.4.2 ASCII コード

ASCII コードとは、ラテン文字や数字などを 7 桁の二進数で表したものである。制御文字を含め 128 文字が収録されている。例えば数字の 0~9 は ASCII コードにおいて 16 進数の 30~39 に対応している。改行を行う場合、3 通りのやり方があり、1 つは CR (キャリッジリターン: ASCII コードで 0x0a)、もう 1 つは (ラインフィード: ASCII コード de0x0d)、最後に両方を使った CR + LF である。今回は受信側に使用する cu コマンドからの制約により CR + LF を使用する。

2.4.3 データ通信

データ通信システムの実装では、FIFO に溜まった XADC の 12 ビットデータを ASCII コードに変換するエンコーダー、エンコーダーで ASCII コードに変換されたデータを送信する送信機、それら 2 つの module と XADC、FIFO を接続するための module の 3 つの module を作成した。データ通信の流れを図 13 に示す。データ通信の流れとしては最初に XADC からの 12bit のデータが FIFO に渡される。XADC は FIFO のデータが満杯になるまでデータを渡し、満杯になったらデータを渡すの停止する。その後、FIFO のデータが読み出され、FIFO のデータが空になったら再びデータを渡し始める。次に、FIFO に溜まっているデータが満杯になったことを検知した場合、エンコーダーは FIFO からデータを読み出し、10 進法表示になるように ASCII コードに変換する。その後、送信機に読み出し有効信号をと一文字分の ASCII コードのデータを送る。送信期が一文字分のデータを送信し、送信終了の信号をエンコーダーに送るとエンコーダーは次の文字を送信期に渡す。エンコーダーは全ての文字を渡し切ると、再び IDLE 状態に戻る。ASCII コードの 7 bit のデータを渡された送信機は、シフトレジスタを用いてシリアルに ASCII コードを出力する。

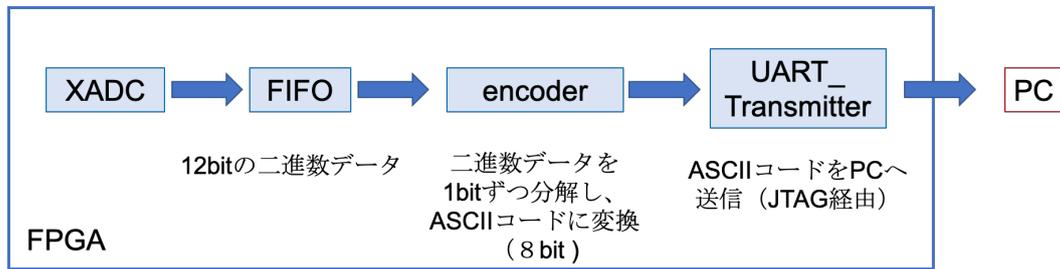


図 13 データ通信の流れ

XADC からのデータ幅は 12bit であり、双極モードで使用するため XADC のデータの範囲を 10 進法で表すと (-2048、2047) である。データ領域として符号も含め 5 桁必要である。また、データの区切りとして改行文字が必要である (cu コマンドでは改行コードは LF + RF のため改行文字は二桁)。従って XADC のデータ 1 つの送信に、ASCII コードで 7 文字、UART で 70 ビット (スタートビットとストップビットを含む) を送信する必要がある。UART の通信速度を 115200 bit/s (UART 通信でよく使われる値) とすると、1 秒間に送信できる XADC のデータは $\cong 1.6$ kHz である。一方、XADC のサンプリングレートは 1 MSPS であり、UART での送信速度を大きく上回っている。そこで、FIFO を使用することでデータを一時的に貯めておき、貯まったデータを UART で送信するようにした。データ通信の概要を図 13 に示す。FIFO の深さは 131072 (最大値) である。エンコーダーと送信機のタイミングチャートをそれぞれ図 14、図 16 に示す。

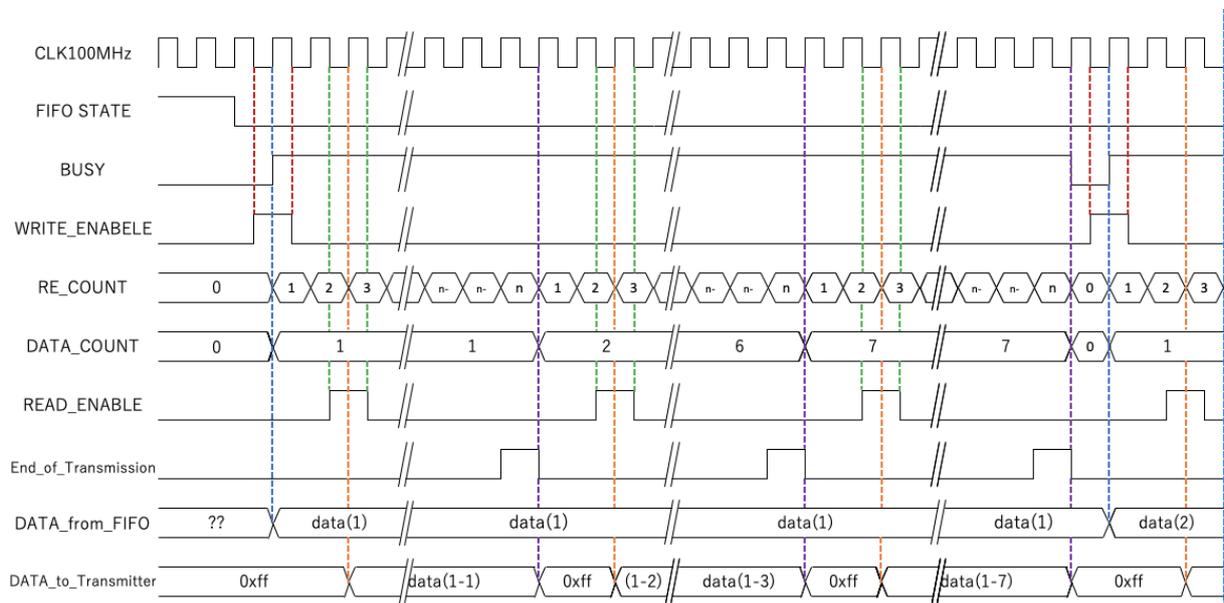


図 14 エンコーダーのタイミングチャート

| | |
|---------------------|--|
| CLK100MHz | 同期用の信号。 |
| WRITE_ENABLE | 送信機の読み出し信号、エンコーダーから出力される |
| DATA_COUNT | 送信データのカウンタを行う。 |
| BAUD_RATE | UART通信のボーレートを決定する。115200 bit/sの時、1/115200で1ビットあたり8.68 μsである。 |
| DATA_Transmission | 送信される信号。 |
| BUSY | 送信機のBUSY信号。送信中にHighを示す。 |
| End_of_Transmission | 送信終了の合図。エンコーダーに出力される |

図 15 エンコーダーの信号の説明

エンコーダーのタイミングチャートについて説明する。エンコーダの入力信号は 100 MHz クロック (CLK100MHz)、FIFO_STATE、END_of_Transmission、DATA_from_FIFO の 4 つの信号があり、出力信号には BUSY、WRITE_ENABLE (以後 WE)、READ_ENABLE (以後 RE)、DATA_to_Transmitter の 4 つがある。RE_COUNT と DATA_COUNT は内部信号であり、エンコーダーの制御に使用する。以下にタイミングチャートの流れを説明する。

1. クロックの立ち下がりで FIFO_STATE、BUSY を監視し、FIFO_STATE が Low かつ BUSY が Low の時、WE を 1 クロック分 High にする。(赤の点線)
2. WE が High になると次のクロックの立ち上がりで BUSY が High に遷移し、RE_COUNT、DATA_COUNT が +1 され、FIFO からデータが渡され (DATA_from_FIFO)、データを ASCII コードに変換する。(青色の点線)
3. その後、RE_COUNT をクロックの立ち上がりの度に +1 し、RE_COUNT が 2 の時に RE を 1 クロックだけ High にする。(緑色の点線)
4. クロックの立ち上がりで RE が High で、かつ DATA_COUNT の値が 1 の時、エンコーダーは ASCII コードの 1 ビット目を送信機に渡す。(橙色の点線)
5. その後、送信機から送信終了の合図 (End_of_Transmission) が送られ、RE_COUNT が 1 にセットされ、DATA_COUNT が +1 される。(紫色の点線)
6. 以後、3～5を繰り返し、最後の送信終了の合図 (End_of_Transmission) で BUSY 信号を Low に戻し、RE_COUNT、DATA_COUNT を 0 にリセットする。

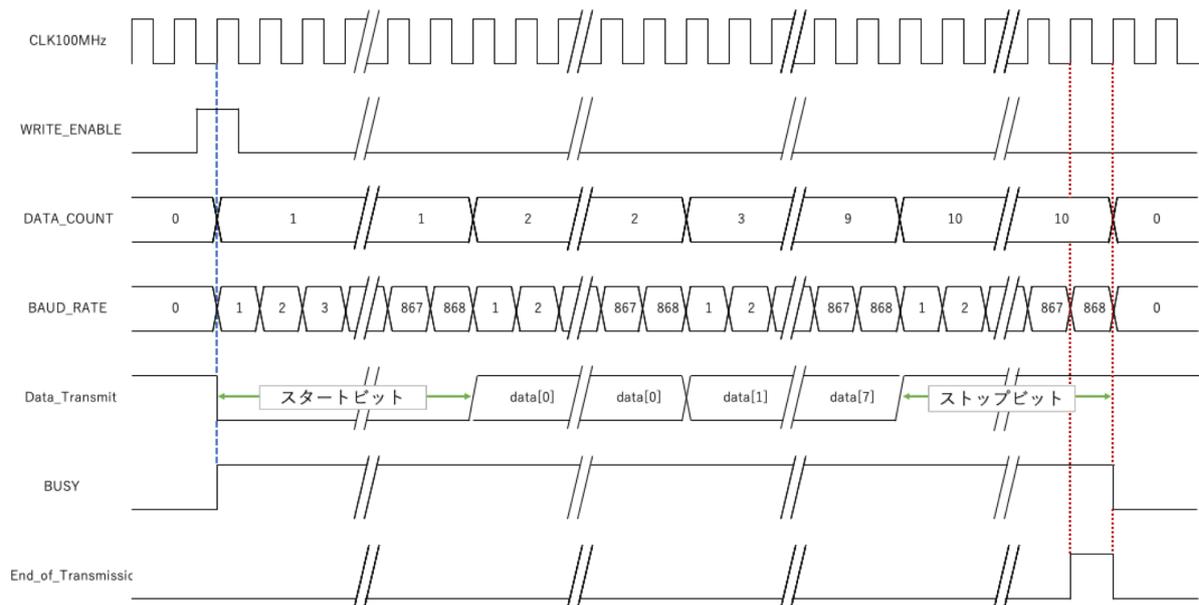


図 16 送信期のタイミングチャート

| | |
|---------------------|-----------------------------------|
| CLK100MHz | 同期用の信号。 |
| FIFO_STATE | FIFOの状態を表す信号読み出し有効の時にLowを示す。 |
| BUSY | エンコーダーのBUSY信号。 |
| WRITE_ENABLE | エンコーダーの書き込み有効信号。 |
| RE_COUNT | エンコーダーの読み出し有効信号の制御用信号。 |
| DATA_COUNT | 送信機に渡すデータをカウントする。 |
| READ_ENABLE | エンコーダーの読み出し有効信号。送信機の書き込み有効信号にもなる。 |
| End_of_Transmission | 送信機から入力され、送信終了を示す。 |
| DATA_from_FIFO | FIFOからのデータ。 |
| DATA_to_Transmitter | 送信機へ出力されるデータ。 |

図 17 送信期の信号の説明

送信機のタイミングチャートについて説明する。送信機の入力信号は 100 MHz クロック (CLK100MHz)、WRITE_ENABLE の 2 つの信号があり、出力信号は DATA_Transmission、BUSY、End_of_Transmission の 3 つの信号がる。DATA_COUNT と BAUD_RATE は制御用の内部信号である。

1. クロックの立ち上がりで WRITE_ENABLE が High の時、BUSY 信号は High に遷移し送信機が動作を開始する。この時 DATA_COUNT と BAUD_RATE を 0 から 1 に変更し、Data_Transmit からスタートビットを送信する。(青の点線)
2. その後、BAUD_RATE はクロックの立ち上がりの度に +1 される。
3. BAUD_RATE が 868 になった時、BAUD_RATE は 1 にリセットされ、DATA_COUNT が +1 される。同時に送信機はデータの 1 ビット目を送信する。
4. 以後、2 ~ 3 を繰り返す、ストップビットまで送信すると、送信機は BUSY を Low に戻し、End_of_Transmission を 1 クロック分だけ High にして動作を終了する。(赤の点線)

2.5 XADC の使用

Artix-7 には 12 ビット、1 MSPS の XADC が 2 つ搭載されている。XADC は制御用とデータ格納用に 16 ビットのレジスタを 128 個持っており、レジスタを読み書きすることで XADC を使用する。128 個のレジスタの内、前半の 64 個のレジスタはステータスレジスタと呼ばれ、XADC の測定データが格納される。後半の 64 個のレジスタはコントロールレジスタと呼ばれ、XADC の動作を定義するために用いられる。FPGA と XADC との間には DRP(Dynamic Reconfiguration Port) と呼ばれるインターフェースが用意されており、FPGA は DRP を通してレジスタの読み書きを行う (図 19)。この DRP を通したレジスタの読み書きは事前に用意されている IP を用いて行う。XADC のポートを図 18 に示す。DRP によるレジスタの読み書きは DCLK と呼ばれる外部から入力されるクロック信号に同期している。本研究では、DCLK に 100 MHz のクロックを使用しており、このクロックに他の module も同期している。

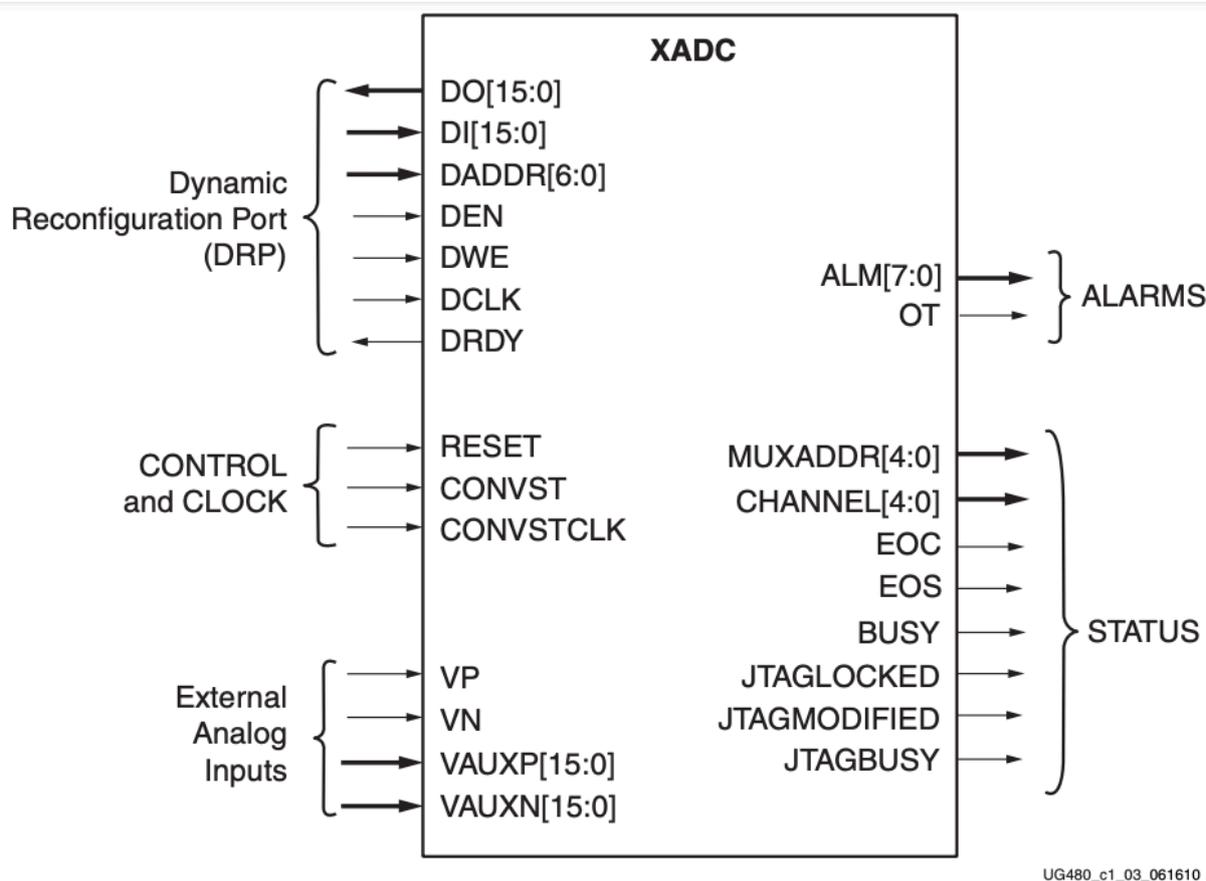
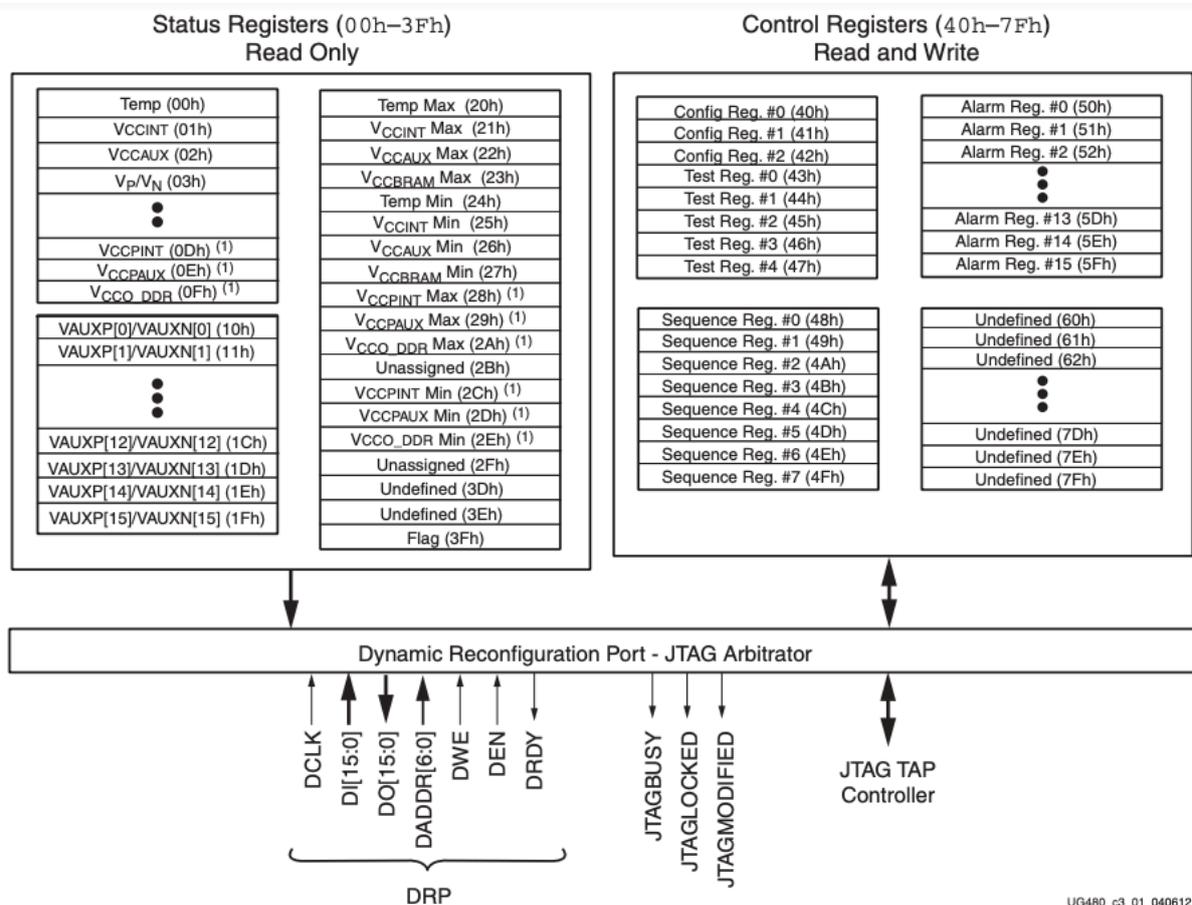


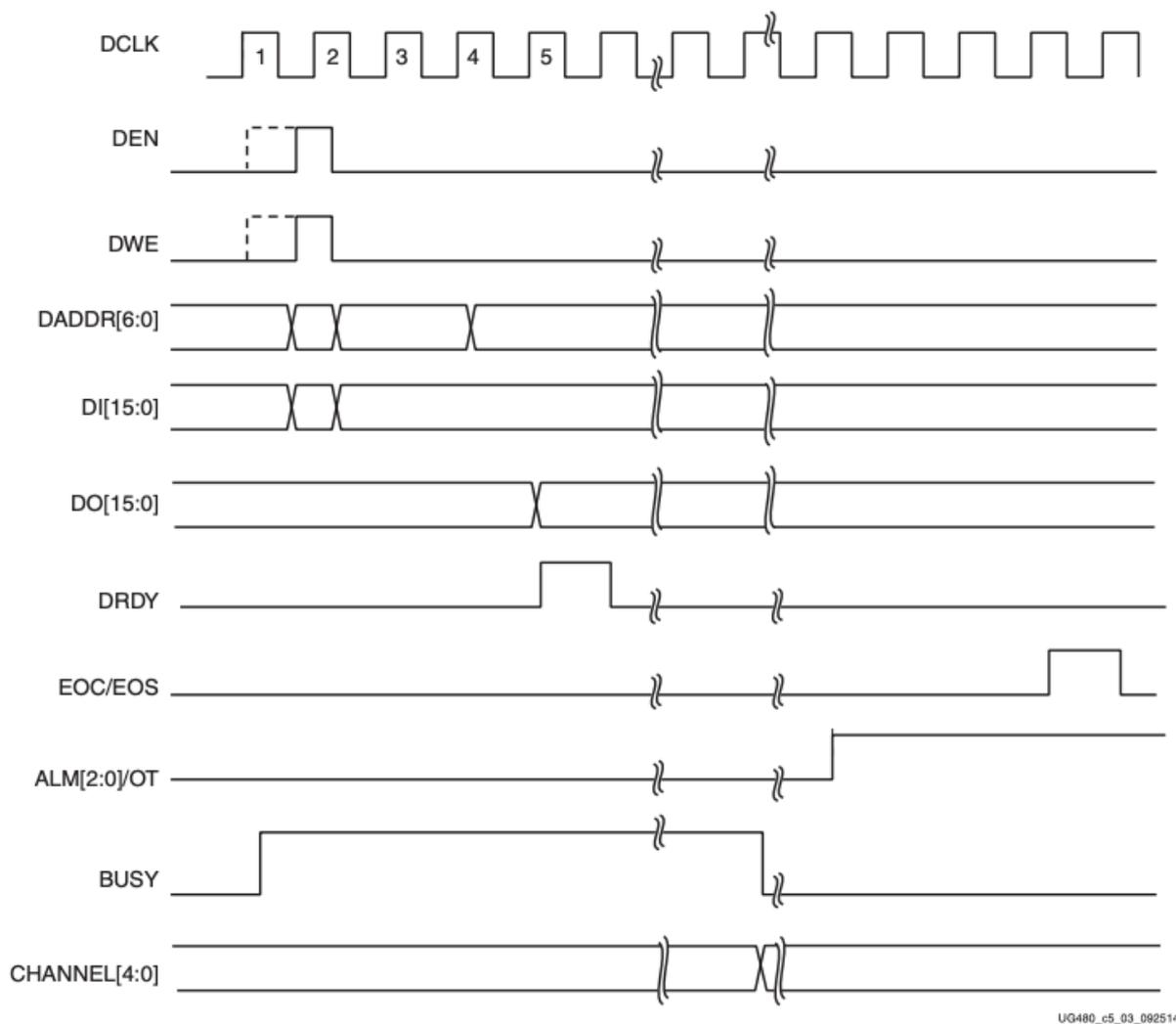
図 18 XADC のポート [3]



UG480_c3_01_040612

図 19 DRP とレジスタ [4]

DRP のタイミングチャートを図 20 に示す。DEN、DWE は外部からの入力信号でレジスタの読み出し、書き込み有効を示しており、このタイミングで DADDR(6:0) で与えられたアドレスのレジスタに DI(15:0) のデータを読み書きする。EOC は XADC で変換されたデータがレジスタに格納されたことを示す信号である。従って、XADC で変換されたデータを取得したい場合、EOC を DEN に接続し、読みたいデータが格納されているレジスタのアドレスを DADDR に入力することで、DO からデータが出力される。ステータスレジスタを読み出した場合、12ビットの測定データは上位12ビットに格納される。



UG480_c5_03_092514

図 20 DRP のタイミングチャート [5]

2.5.1 XADC の較正

XADC の線形性を確認し、伝達関数を導く。XADC のダイナミックレンジは 1 V で、差動モードで使用するため V_p 端子の V_n 端子に対する電圧範囲は -500 mV から 500 mV である。定電圧として、安定した電圧出力を持つ乾電池を抵抗で分圧したものを用い、端子間に -400 mV、-292 mV、-198 mV、-92 mV、0 mV、92 mV、198 mV、292 mV、400 mV の電圧をかけた。かけた電圧に対しそれぞれ 35000 回分の XADC のデータを取得しヒストグラム (図 21) を作成した。作成したヒストグラムをガウシアンでフィッティングした時の平均値をその電圧の XADC の値として採用した。XADC の精度についてヒストグラムの標準偏差を図 22 に示す。これを見ると標準偏差は 0.3 mV 程度であり、計測する電圧によって大きく変化しないことがわかる。

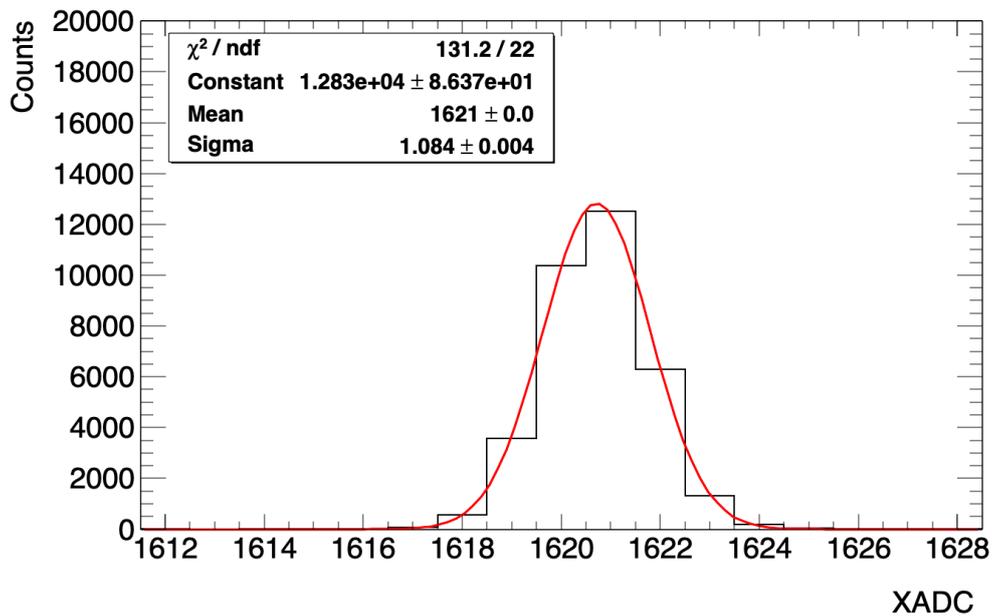


図 21 400mV の電圧をかけた時の XADC の値

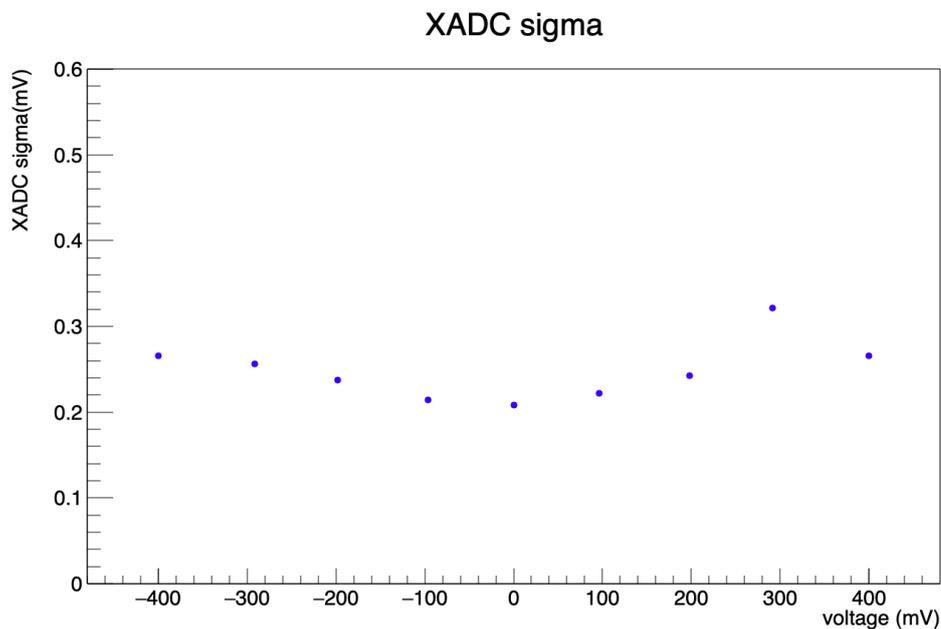


図 22 XADC の標準偏差

かけた電圧と XADC の値から線形回帰 (図 23) を行い、伝達関数 (式 1) を得た。この式から XADC の最小分解能は 0.2452 mV、入力電圧範囲は約-502.29 mV 501.89 mV だとわかる。以降、この伝達関数を用いて XADC の値を電圧に変換する。

$$voltage(mV) = (0.2452 \pm 0.0003)XADC + (0.08 \pm 0.33) \quad (1)$$

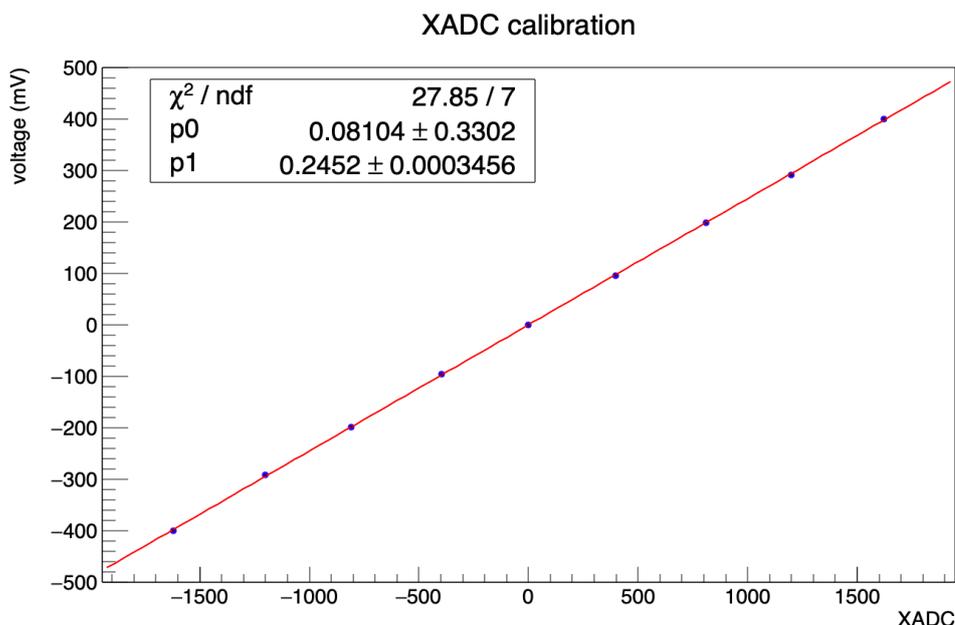
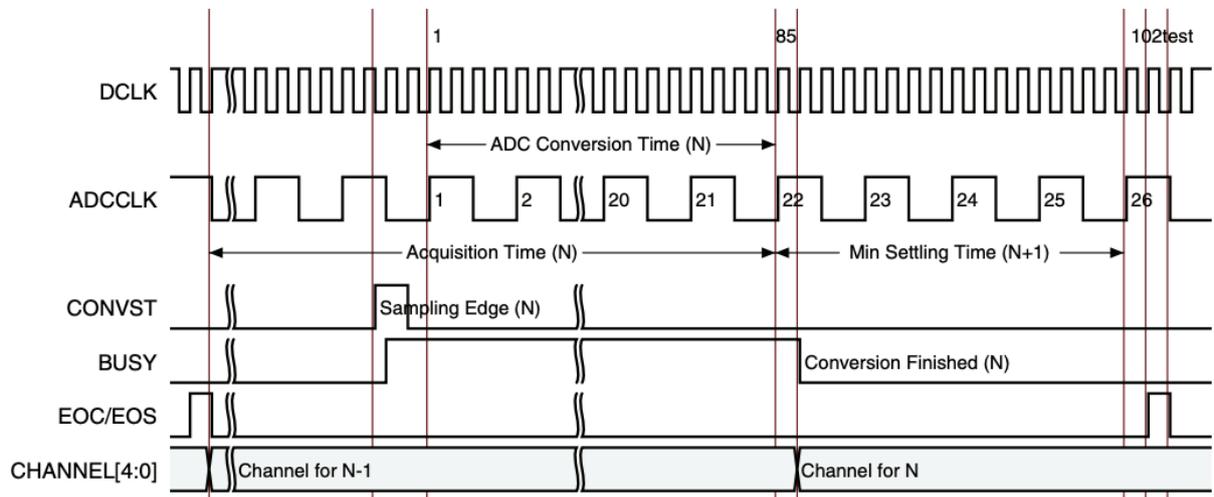


図 23 電圧と XADC の値の関係

2.6 センサ読み出しシステム

2.6.1 XADC とセンサの VIDEO 信号の同期

先述したデータ通信システムにより、XADC のデータを PC に取り込めるようになった。しかし、これまでのデータ通信では XADC は連続サンプリングモードで動作させており、XADC は AD 変換が終わるとすぐに次の変換を開始する。一方、ラインセンサからの VIDEO 信号は図 4 に示すように、周期が 1 MHz のクロックで 4 クロック分 (0.25 MHz) である。XADC のサンプリングレートは 1 MSPS であるため、連続サンプリングモードでは不必要な AD 変換を行うことになる。XADC をイベントサンプリングモードで動作させることで、センサの VIDEO 信号のタイミングに合わせて AD 変換を実行させる。イベントサンプリングモードでの XADC のタイミングチャートを図 24 に示す。イベントモードでは convst 信号の立ち上がりを検出すると、その時のアナログ入力をサンプリングし、その後 AD 変換を行う。従って、VIDEO 信号の出力期間に一回だけ convst が High になるようにする。これには、センサ制御信号の生成の為に用意したカウンターを用いる。センサからの VIDEO 信号は 1 MHz クロックの 2 クロック分出力されるが、推奨取り込み期間は後半の 1 クロックの間である。また、convst 信号は 100 MHz の DCLK に同期させる必要があり、XADC の双極モードでの最低取得時間が 300 nsec[6] であることから、convst 信号のタイミングは 1 MHz のクロックの 21 クロック目から 4 クロックおきで、かつ 100 MHz クロックのカウンターが 30 以降である必要がある。本研究ではカウンターが 48 のときに 100 MHz クロックで 1 クロックだけ出力する。(図 25)



UG480_c5_02_092414

図 24 イベントサンプリングモードのタイミング [7]

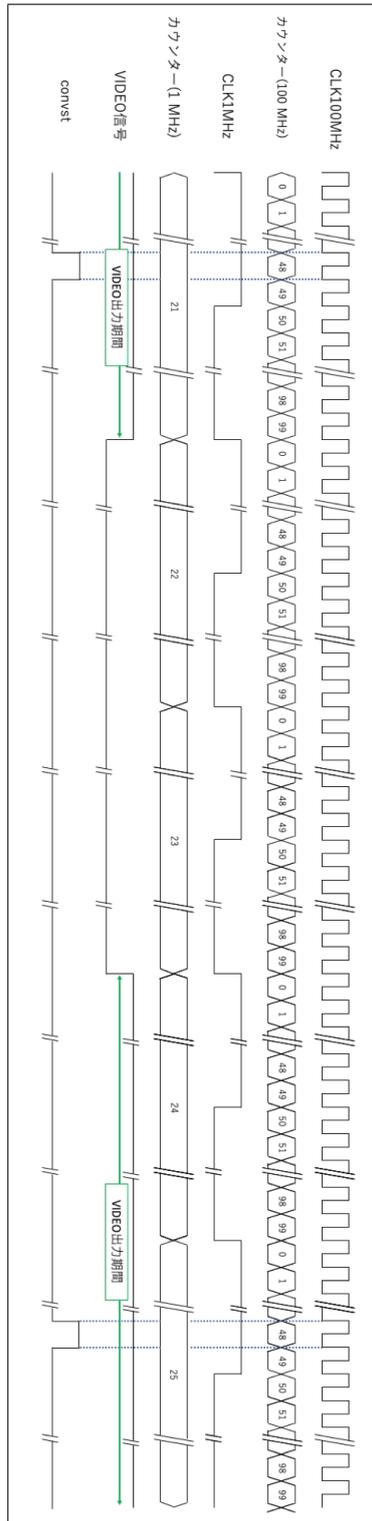


図 25 convst 信号のタイミング

2.6.2 センサのチャンネル情報

使用するラインセンサはフォトダイオードが一行に並んだものであり、256 ch で 1 セットのデータであるため、XADC の測定データに加えてチャンネル情報も取得したい。そこで、XADC の測定データが出力された時の 1 MHz のカウンターを測定データと同時に FIFO に書き込むことでチャンネルデータを得る。

2.6.3 FIFO へのデータの書き込みについて

XADC の較正で使ったデータ通信システムは FIFO の深さを 131072 とし、満杯になった場合は一度書き込みを中止してデータが空になったらデータの書き込みを再開する仕組みである。ラインセンサからのデータを書き込む場合、チャンネル情報が含まれるため、書き込みを中止する時は 256 ch まで書いてから、書き込みを再開する時は 1 ch 目のデータが来るのを待ってから書き込みを再開する必要がある。書き込みの中止に関しては、FIFO の深さが 256 の整数倍あるため FIFO が満杯になるのを待てば良い。書き込みの再開に関しては、FIFO のデータが空になった後、256 ch のデータが出力されたのを確認した後に書き込みを再開する様にした。

2.6.4 UART でのデータ読出の速度について

センサ読み出しシステムの読み出しの速度について、1 ch の測定データはチャンネル情報を含めて 23 ビットであり、ASCII コード 12 文字分（チャンネル情報 4 文字、測定データ 5 文字、データの区切り、改行 2 文字）である。256 ch 分のデータは ASCII 文字で 3072 文字で、30720 ビットであるから、UART で連続に送信したと仮定した場合、115200 bit/s の通信速度でデータレートは 3.75 /s である。

3 結果

3.1 暗電流

暗電流状態でのシリコンラインセンサからの VIDEO 信号を観測した。セットアップを図 26 に示す。

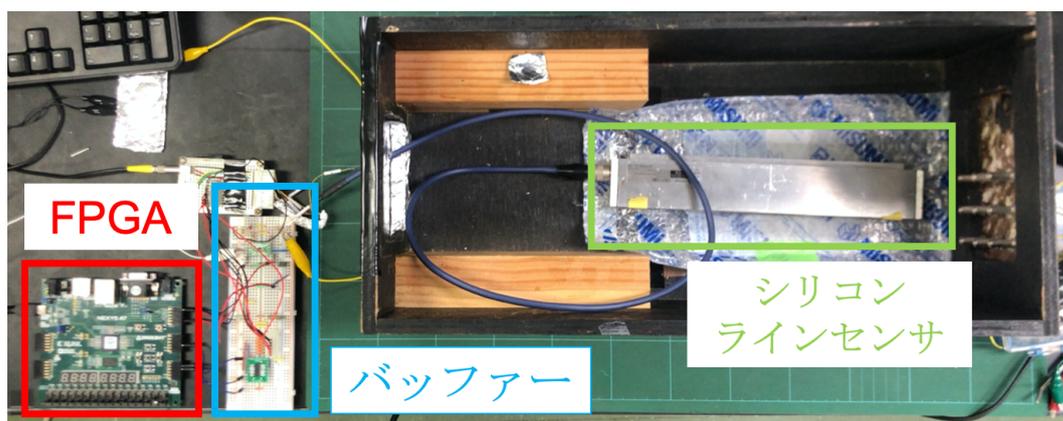


図 26 FPGA とシリコンラインセンサのセットアップ

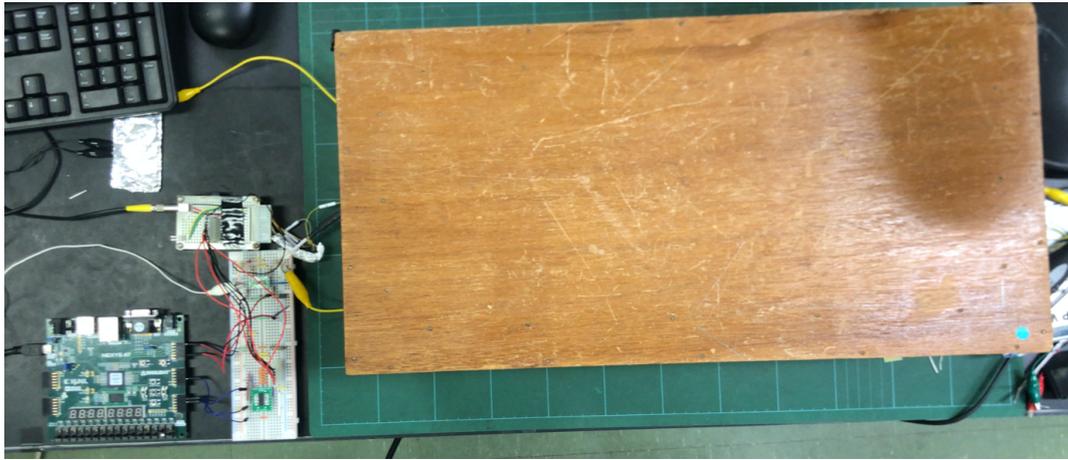


図 27 暗箱を閉めた様子

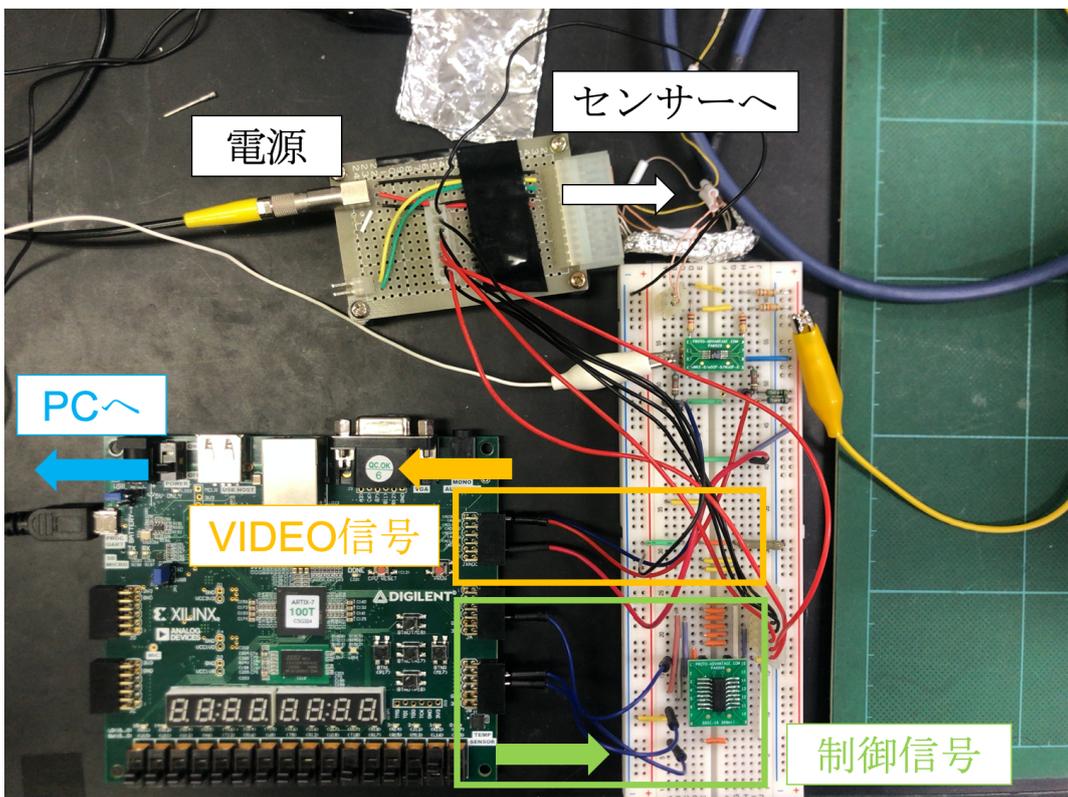


図 28 FPGA 部分

暗電流状態で、シリコンラインセンサからの VIDEO 信号の ADC 分布を図 29 に示す。図 29 は蓄積時間が 89.99 ms の時の 256 ch 目の VIDEO 信号を 5120 回測定した時のヒストグラムである。すべてのチャンネルで、図 29 の様なヒストグラムを作成し、平均値を VIDEO 信号の値としてグラフにしたものが図 30 である。横軸はチャンネルで 1 チャンネルから 256 チャンネルまでである。128 チャンネルと 129 チャンネルの間

で値が不連続に変化するが、これはセンサの構造によるものである。センサの 256 個のフォトダイオードは 128 個が 1 つの基板上にあり、それを 2 つ繋げて用いているため 128 ch と 129 ch の間で不連続となる。

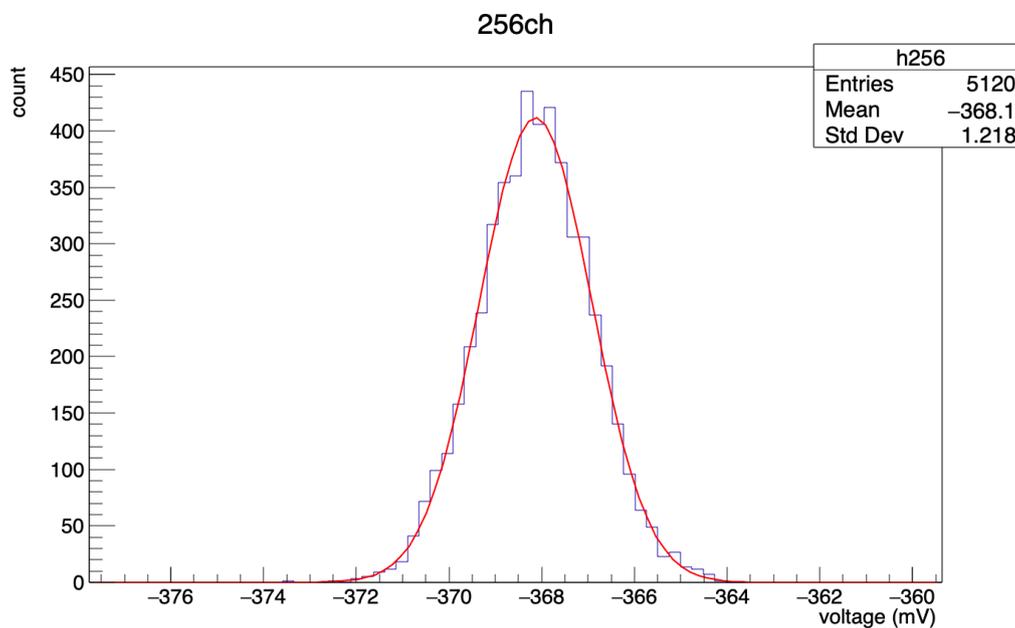


図 29 暗電流における 256 ch のヒストグラム

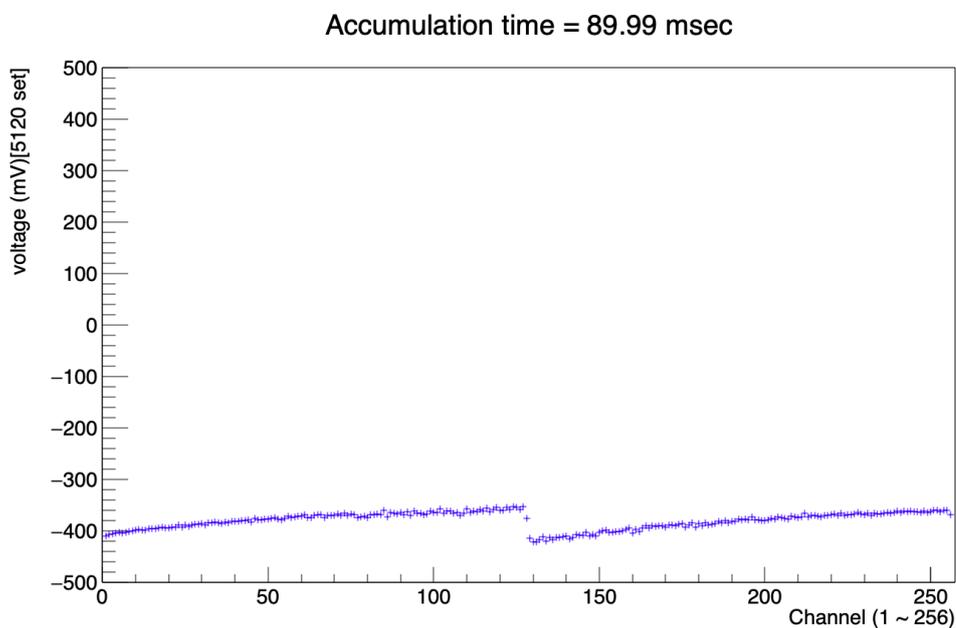


図 30 暗電流状態でのセンサからの VIDEO 信号 (5120 セットの平均)

図 31 と図 32 は 5120 回分の VIDEO 信号をプロットしたものである。図 31 と図 32 を見ると VIDEO 信

号のばらつきは 10 mV 程度であり、全てのチャンネルでおおよそ一定である。チャンネルごとにヒストグラムを作成し、ガウス関数でフィッティングした時の標準偏差 (図 33) は 10 mV 程度である。

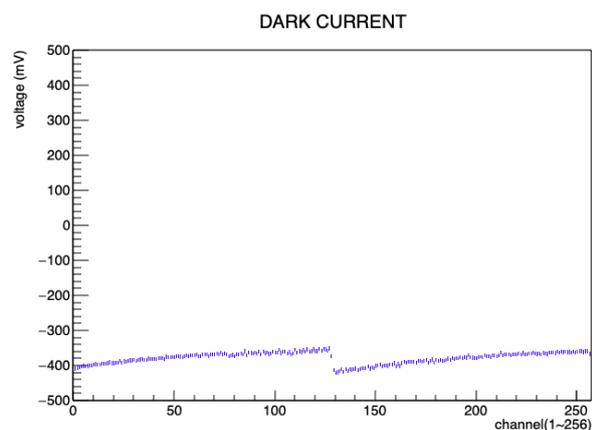


図 31 暗電流状態でのセンサからの VIDEO 信号のプロット

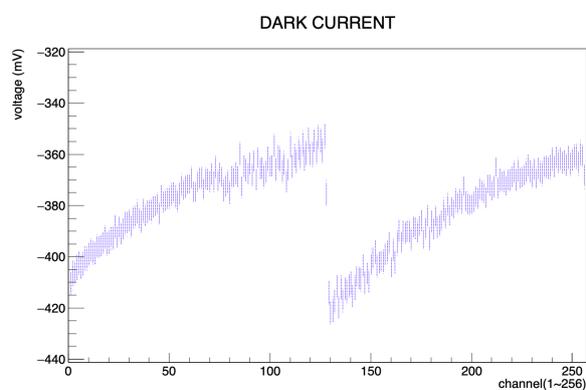


図 32 暗電流状態でのセンサからの VIDEO 信号のプロット (拡大)

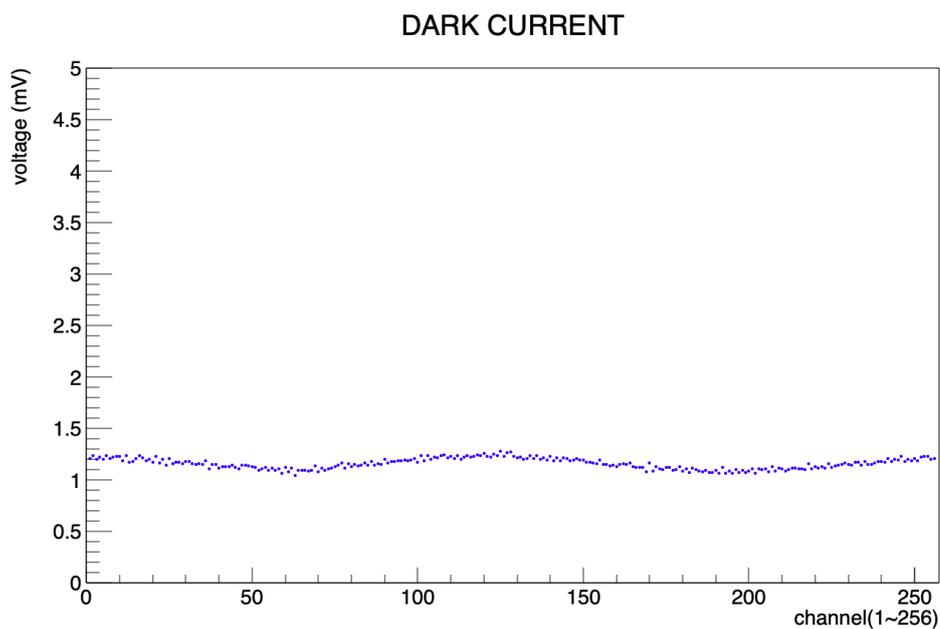


図 33 暗電流におけるチャンネルごとの標準偏差

3.2 センサによる光の検出

実際にセンサに光を当ててセンサの応答を確認する。センサの周りは金属板で覆われているため、ねじ穴 (図 34) から光を入れて確認する。図 35 はねじ穴から光を入れた状態での VIDEO 信号をプロットしたもの

で 5120 セット分のデータを平均したものである。図 36 は 5120 セット分のデータを重ね書きしたものである。これを見るとねじ穴付近のチャンネルが反応していることが確認できる。205 ch 付近に小さなピークが見えるが、これはセンサヘッド内部の基板や配線などで影ができるためと考えられる。標準偏差は図 37 の様になり、VIDEO 信号の電圧が大きくなると標準偏差も大きくなる。



図 34 ラインセンサ

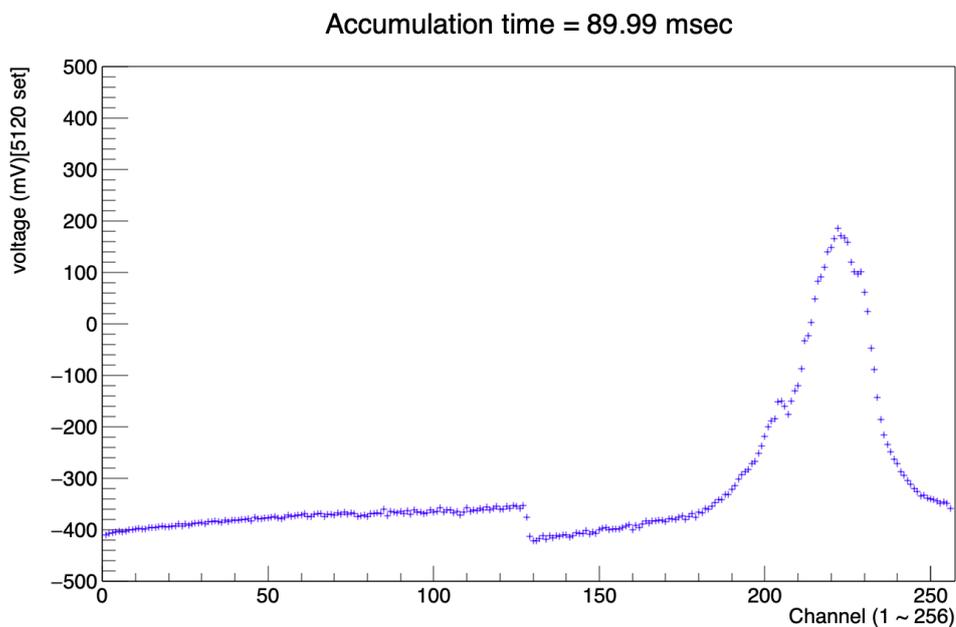


図 35 ねじ穴から光を入れた時のセンサの応答 (5120 セットの平均)

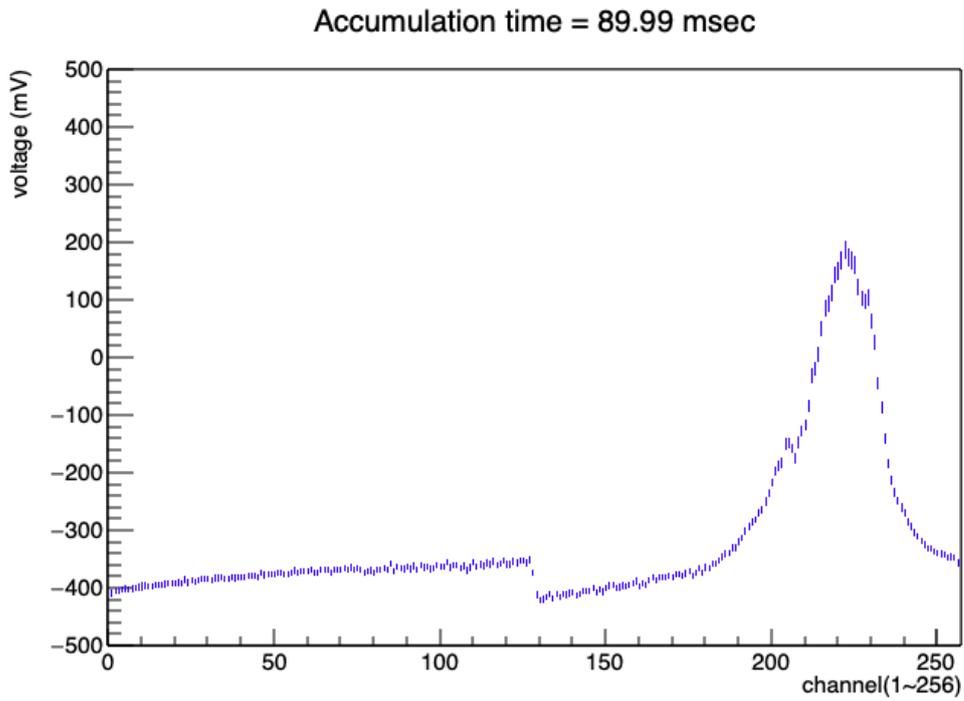


図 36 光を入れた状態での 5120 回分のプロット

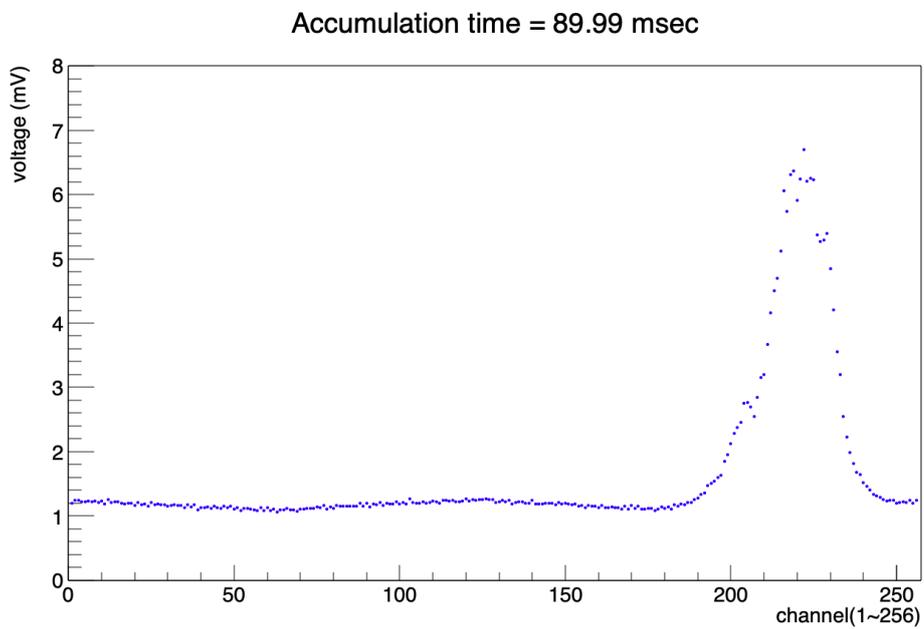


図 37 ねじ穴から光を入れた時のチャンネルごとの標準偏差

光ありの状態のヒストグラムから暗電流のヒストグラムを引いたものが図 38 である。

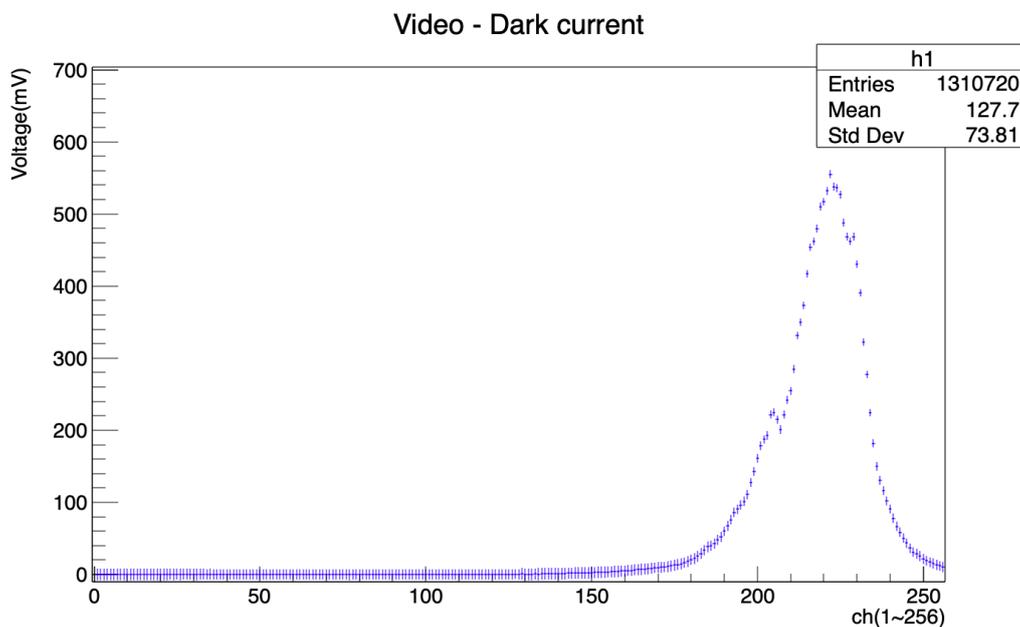


図 38 光を当てた時の暗電流によるバックグラウンドを引いた時のヒストグラム

また、蓄積時間によって VIDEO 出力がどの様に変化するか観察した。図 39 は上から蓄積時間が 89.99 msec、79.99 msec、69.99 msec、59.99 msec、49.99 msec、39.99 msec、29.99 msec、19.99 msec である。これより、蓄積時間に応じてセンサの VIDEO 出力も大きくなることがわかった。それぞれの蓄積時間でのヒストグラムをチャンネルで積分し、蓄積時間を横軸に、積分した値を縦軸にしてプロットした (図 40)。図 40 より、蓄積時間と VIDEO 信号の出力は比例していると言える。

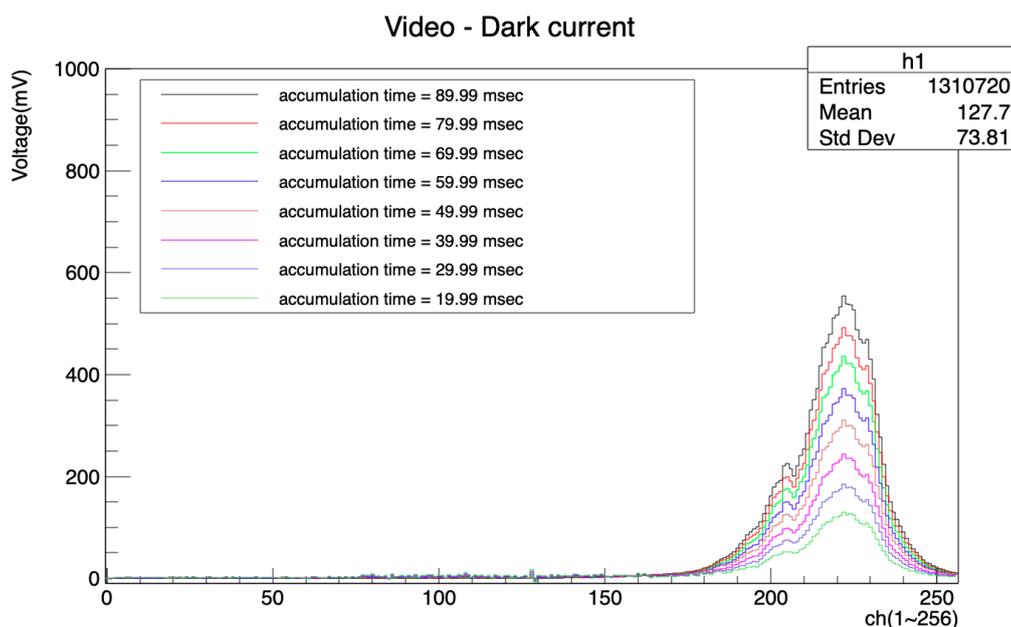


図 39 蓄積時間の違いによる VIDEO 信号の変化

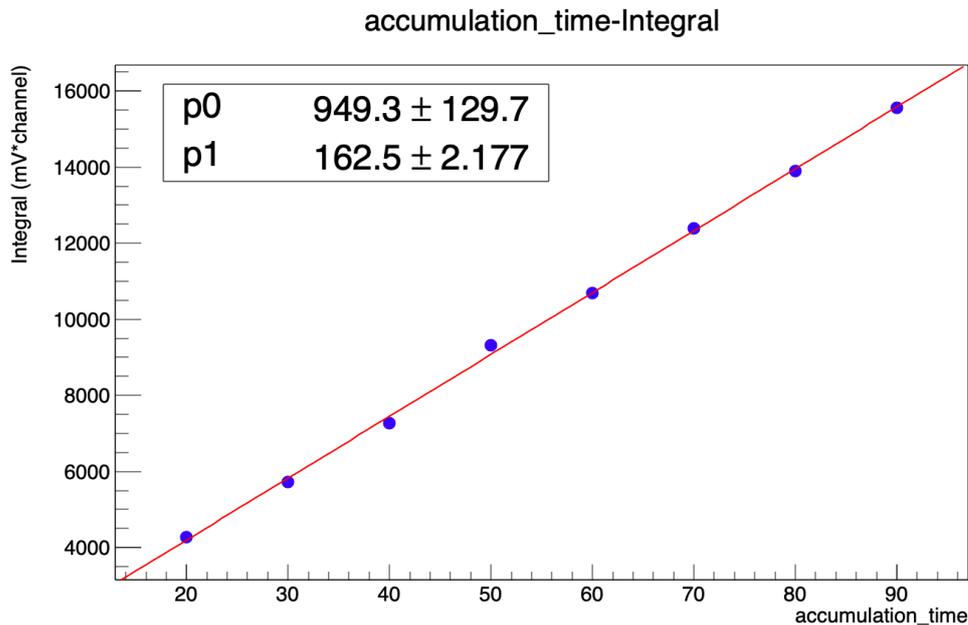


図 40 蓄積時間とヒストグラムの面積

4 今後の課題

今回の研究で FPGA を用いてシリコンラインセンサを駆動し、センサからの信号を AD 変換、UART を用いることで PC にデータを送信することができた。しかし、現状の UART では PC へのデータ送信においてデータレートが 3.75 /s と、センサのデータレート（～1 k/s）に比べ非常に低い。さらに、センサは 4 MHz のクロックでも動作させることができ、その場合センサのデータレートは～4 k/s まで上昇することになる。シリコンラインセンサのデータ処理の高速化において、データ通信の高速化は必須である。現在使用している FPGA ボードの Nexys A7 には 100Mbps のイーサネットがサポートされており、より高速な通信を実現することが可能である。イーサネットを使用するにあたって、MAC などのデータリンク層の処理を FPGA で行う必要がある。現段階では、データリンク層などの処理に Microblaze(Vivado で使用できる FPGA 上で構成する CPU) を使用する方法、データリンク層の処理を FPGA で直接行う方法が考えられる。前者の CPU を使う方法では、Vivado に加えて Vivado SDK (Microblaze をプログラムするためのソフトウェア開発キット) を使う必要も出てくるが、サンプルプログラムがいくつか用意されている模様である。後者の CPU を使わずに直接 FPGA で処理を行う方法では SDK などを使わずに済むが、一から設計しなえればならぬ。

5 まとめ

今回の研究では、FPGA を用いたセンサ制御信号の生成、及び FPGA のオンチップ ADC を用いたセンサからの VIDEO 信号の処理、PC との UART によるデータ通信を実装し、センサ制御システムを開発した。開発したセンサ制御システムを用いてシリコンラインセンサの動作を確認した。今後の課題として、イーサネットを用いたデータ通信の高速化、さらなるデジタル回路の集約化を行う必要がある。

謝辞

本研究を進めるにあたり、お世話になった多くの方々に感謝いたします。山口頼人准教授には研究を進める上で様々な助言をいただき、また解析ミーティングの発表の指導などもしていただきました。サイエナジー社には X 線センサをお貸しいただきました。サイエナジー社の阿部圭一様には、センサについて細かいところまで教えていただきました。長崎総合科学大学の大山健様、浜垣秀樹様には、研究を進める上で様々な助言をいただきました。志垣賢太教授にはミーティングなどで様々な質問をしていただきました。本間謙介助教には前期のラボエクササイズで物理実験の基礎を教わりました。三好隆博助教にはセミナーを担当していただきました。八野哲助教にはミーティングで様々な質問をいただきました。山川さんには研究に関することから研究以外のことまでお世話になりました。木村さんには PC の手配などでお世話になりました。研究室の先輩方、同期のみなさんも仲良くしてくれありがとうございます。最後に、今まで陰ながら支えてくれた家族に感謝します。

参考文献

- [1] https://www.hamamatsu.com/resources/pdf/ssd/10_handbook.pdf p.252
- [2] http://accwww2.kek.jp/oho/oho16/OHO16_txt/04_Uno_syouji.pdf 2.2 イオン化 19 行目
- [3] 7 シリーズ FPGA および Zynq-7000 All Programmable SoC XADC デュアル 12 ビット 1 MSPS アナログ-デジタルコンバーター ユーザガイド p.14
- [4] 7 シリーズ FPGA および Zynq-7000 All Programmable SoC XADC デュアル 12 ビット 1 MSPS アナログ-デジタルコンバーター ユーザガイド p.32
- [5] 7 シリーズ FPGA および Zynq-7000 All Programmable SoC XADC デュアル 12 ビット 1 MSPS アナログ-デジタルコンバーター ユーザガイド p.71
- [6] 7 シリーズ FPGA および Zynq-7000 All Programmable SoC XADC デュアル 12 ビット 1 MSPS アナログ-デジタルコンバーター ユーザガイド p.26(式 2-4)
- [7] 7 シリーズ FPGA および Zynq-7000 All Programmable SoC XADC デュアル 12 ビット 1 MSPS アナログ-デジタルコンバーター ユーザガイド p.69
- [8] Open-It FPGA トレーニングコース (入門編)
- [9] <https://ameblo.jp/middleisland88/entry-11558478782.html> はいてくどかたのヒトリゴト
- [10] <http://www.darwin.esys.tsukuba.ac.jp/home/ohyou/verilog/about>