2021 年度 卒業論文

シリコン検出器の Gbps 読出に向けたデータ伝送システム開発

2022 年 2 月 8 日

指導教員　山口頼人　准教授
主査　山口頼人　准教授
副査　栗木雅夫　教授

広島大学
所属 クォーク物理学研究室

学籍番号　B182160
添田拓

概要

物理実験においてデータ読出の高速化は収集可能データ量に直結する重要な要素である。加速器の高輝度化により増大したデータ量に対応するため、読出回路では高速かつ大容量のデータ伝送システムやリアルタイムデータ処理などが重要である。次世代の読出システム開発において SoC-FPGA チップが注目されている。SoC-FPGA チップは CPU が組み込まれた FPGA チップであるので、CPU 制御によるギガビットイーサネットでの伝送システムや、CPU を用いたデータ圧縮などのデータ処理機能が実装できる。これらにより検出器読出技術の高度化が実現すると期待している。本研究では SoC-FPGA チップ搭載の開発ボードを用いてシリコンセンサの次世代読出回路の開発を行った。制御信号を生成する回路、センサ信号を A/D 変換する回路、CPU を用いたリアルタイムデータ処理、ギガビットイーサネットを用いたデータ伝送回路を集約ならびにセンサ信号の再構成を目指した。このうち、制御信号生成回路、A ／ D 変換回路の設計及び実装に達成した。CPU 機能を利用したシステム開発が今後の課題である。

目 次

図 目 次

第1章　序論

1.1　読出システムの高度化

加速器の高エネルギー化・高輝度化により、1 衝突事象あたりのデータ量や衝突レートは増加している。これにより 1 秒あたりの衝突事象数は劇的に増加し、読出されるデータ量は膨大になる。よって読出システムでは高速かつ大容量のデータ伝送システムやデータ圧縮などのリアルタイムデータ処理の要素が必要である。全ての検出器は、これを受けて読出技術の高度化を必要としている。これらの要素を実現するためにギガビットイーサネット、CPU 組込 FPGA チップが有望である。これらの技術を搭載しているチップとして、SoC-FPGA チップが次世代の読出回路構成の主流になると期待されている。

本研究ではシリコン検出器の読出に取り組む。今日の原子核・素粒子実験においてシリコン検出器は必要不可欠なものとなりつつある。放射線や荷電粒子が半導体中を通過すると、電離作用により電子-正孔対が生成され、電流が流れる。これにより放射線・荷電粒子を検出することができ、シリコン検出器と呼ばれる。シリコン検出器は電子-正孔対を形成するのに必要なエネルギーは約 3.6 eV [1] と、気体のイオン化に必要なエネルギーの約 10 分の 1 と小さく [2]、エネルギー分解能に優れている。半導体技術の進歩によりピクセルサイズを小さくできるので、高い位置分解能を実現している。これにより、シリコンピクセル検出器は位置検出器として機能する。単位面積あたりのピクセル数が増え、データ量も増加する。従ってシリコン検出器の読出システムの高度化は重要さを増している。

1.2　読出回路構成

読出回路を構成する要素を簡潔に示す。本回路の中心となるのが SoC-FPGA チップである。次にデータの A/D 変換のために ADC が必要である。最後にデータを外部機器に伝送するためにイーサネットが必要である。

1.2.1　SoC-FPGA チップ

FPGA (Field Programmable Gate Array) の最大の特徴は、デジタル回路を柔軟に変更できることである。これにより多種多様な機能を設計・実装が可能であり、その汎用性が注目されている。従来、読出回路を構成する際には ASIC (Application Specific Integrated Circuit) が主流であった。ASIC には一つの目的に向けた機能特化という利点のほかに、その実装までに時間やコストがかかる点や実装後には構成回路を修正できないという物理実験には大きいデメリットがある。そのため現在は読出回路に FPGA が使用されることが一般的になっている。

SoC とは System on Chip の略である。SoC-FPGA は FPGA に CPU やメモリなどを組み込んだチップである。CPU 制御により今までは FPGA と独立に繋げていた A/D 変換回路やイーサネットなどの回路を 1 つに集約できる。FPGA が得意とする高速並列処理と CPU が得意とする

複雑なデータ処理を同一回路内で実現でき、リアルタイムデータ処理などの高度で高速な処理を可能にする [4]。

1.2.2 ADC (Analog to Digital Converter)

連続的に変化するアナログ信号をデジタル信号に A/D 変換 (アナログ/デジタル変換) するモジュールである。ADC は時間や電圧・電流値 (=アナログ値) をどれだけ細かく変換できるかの指標である分解能を持つ。これが A/D 変換の再現度である解像度を決めている。

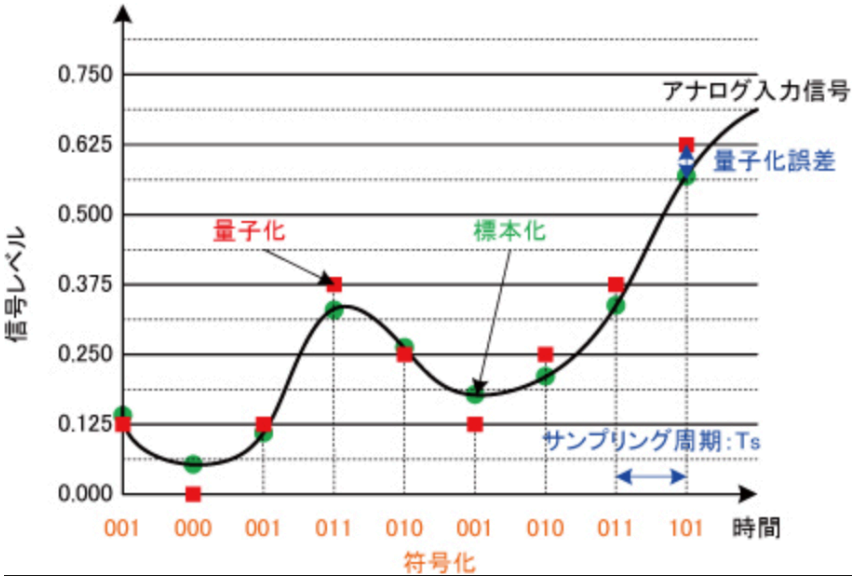


図 1.1: A/D 変換の過程 [5]

図 1.1 は A/D 変換の過程である。変換には標本化・量子化・符号化の 3 段階がある。標本化とは、アナログ信号の振幅値を離散的な周期で区切る過程である。これをサンプリングといい、区切った周波数をサンプリング周波数またはサンプリングレートという。量子化とは、サンプリングされた振幅を離散的なデジタル値に近似する過程である。この時、アナログ値とデジタル値の間に必ず誤差が生じる。これを量子化誤差という。この近似幅を決定しているのがビット分解能である。サンプリングレート・ビット分解能が高い方がよりアナログ信号に忠実なデジタル変換が可能である。符号化とは、量子化によって離散的に変換された振幅値を 2 進数で表現することである。以上の行程を経て連続的なアナログ信号は離散的なデジタル信号に変換される。

1.3 本研究の目的

本研究では、SoC-FPGA チップを用いて

- 制御信号生成
- A/D 変換
- データ処理
- データ伝送

の 4 つの機能を 1 つのデジタル回路に集約したシリコンセンサのデータ読出システムを開発することを目指した。図 1.2 は読出システムの概要である。必要な 4 つの機能を持つ回路が集約していることがわかる。

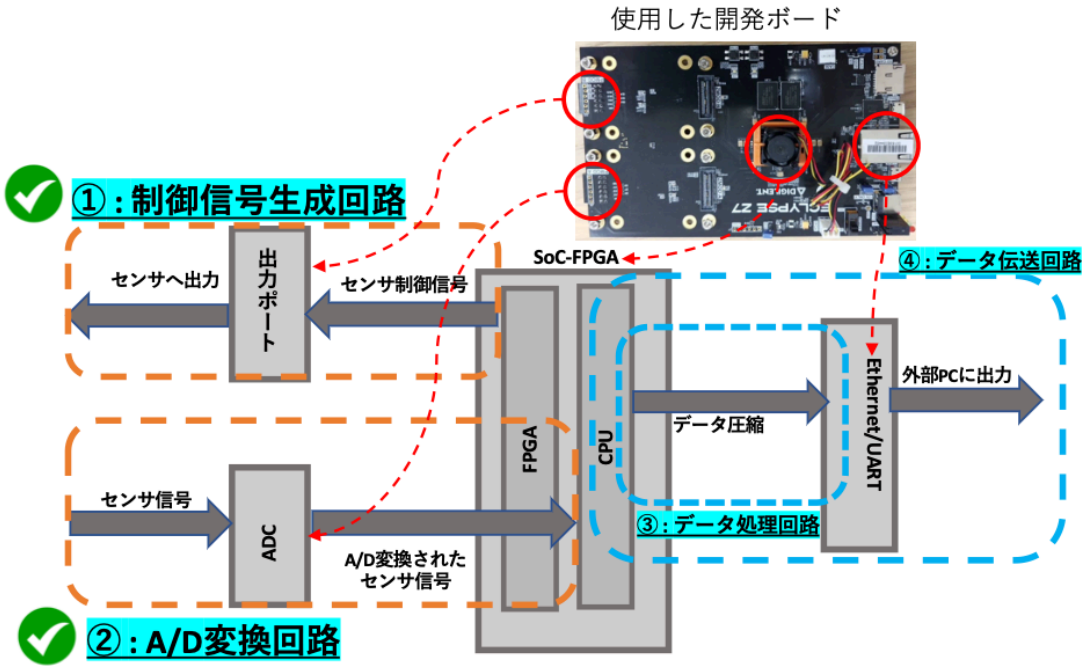


図 1.2: 読出システムの概要

第2章　開発環境

2.1　ハードウェア

2.1.1　FPGA チップ

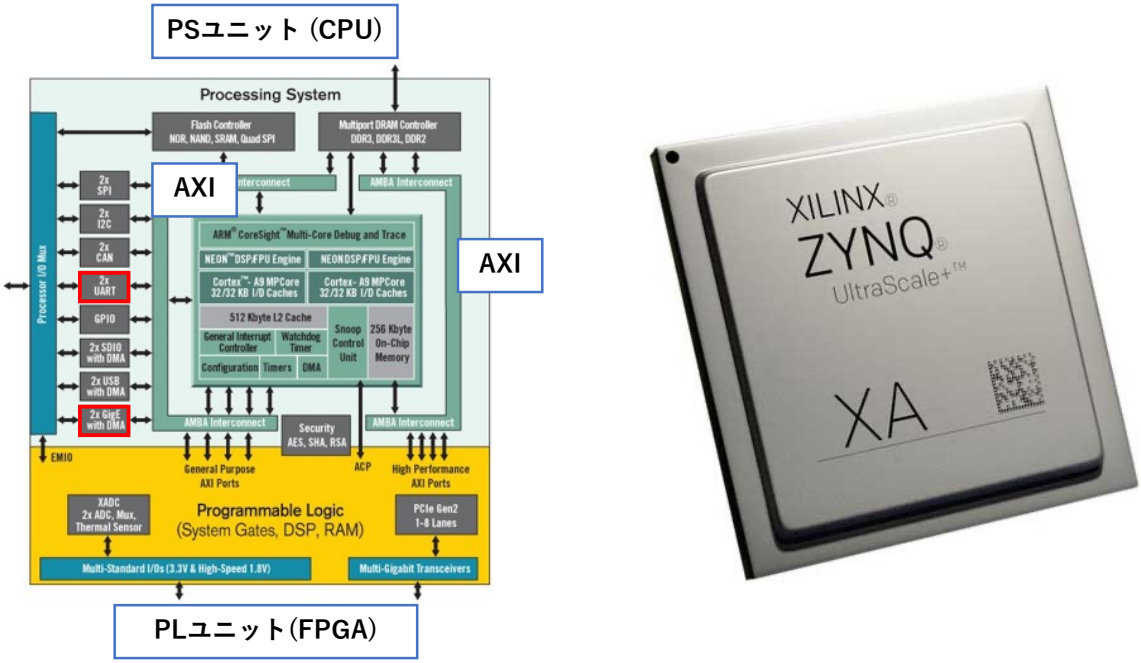


図 2.1: Zynq-7000 のブロック図 [7]・外観 [6]

Xilinx 社の Zynq-7000 を用いた。図 2.1 は Zynq-7000 の内部ブロック図と外観を示している。このチップは PS (Proseccing System) と PL (Programmable Logic) の 2 つのユニットから構成されている。PS/PL ユニットはそれぞれ CPU/FPGA に対応している。各ユニットは AXI (Advanced eXtensible Interdace) というインターフェースを通して通信する。CPU 制御によりギガビットイーサネット (GigE) や UART などの通信プロトコルが利用できる。

2.1.2　開発ボード

開発ボードとして Digilent 社の Eclypse Z7 (図 2.2) を使用した。チップセットに Zynq-7000 を、その他様々な I/O コネクタを搭載している。今回は制御信号の出力と外付け ADC の接続に汎用 I/O 規格である Pmod コネクタを、データ伝送のテスト用に UART ポートを使用した。

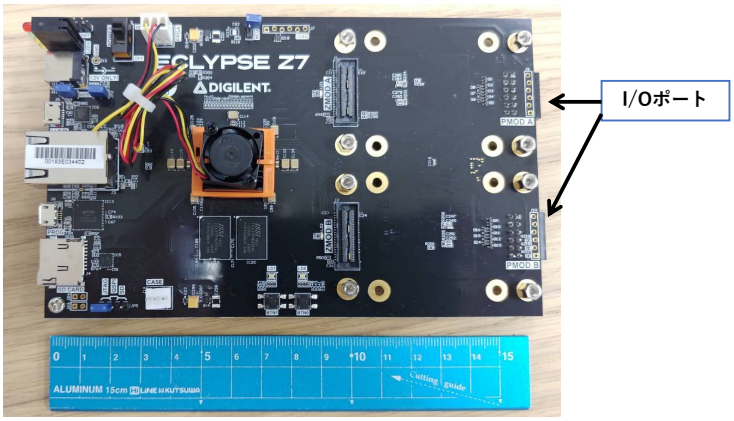


図 2.2: Eclypse Z7

2.1.3 ADC

Digilent 社が提供している Pmod 規格の外付け ADC モジュールである Pmod AD1 を用いた。A/D 変換チップには AD7476A を搭載している。これは 2 ch の同時変換が可能で、12 bit の解像度、1 MSPS のサンプリングレートを持つ。Eclypse Z7 の汎用 I/O (図 2.2 参照) である Pmod のうち、片方に接続して使用している。

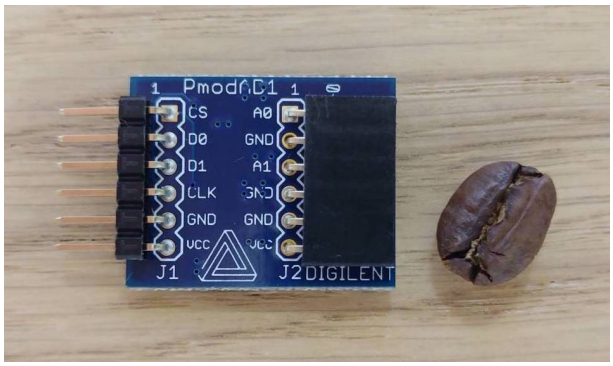


図 2.3: Pmod AD1

2.2 ソフトウェア

システム開発には Xilinx 社が提供しているツールである Vivado、Xilinx SDK、PetaLinux を用いた。Vivado では Verilog というハードウェア記述言語を用いて PL ユニットの機能設計が可能である。SDK (Software Development Kit) は、PS ユニットに C 言語や C++言語ベースのプログラムを実装することが可能である。PetaLinux は Xilinx 社製ハードウェアの公式 OS である。Zynq-7000 の CPU 上で Linux を起動することができ、SDK で実装したプログラムと組み合わせてデータをリアルタイムに処理することが可能になる。

2.3 システム開発の流れ

具体的な開発の流れを以下に示す。図 2.4 は開発の流れを簡潔にまとめたものである。必須な過程は実線で、必ずしも必須ではない過程は破線で表現した。

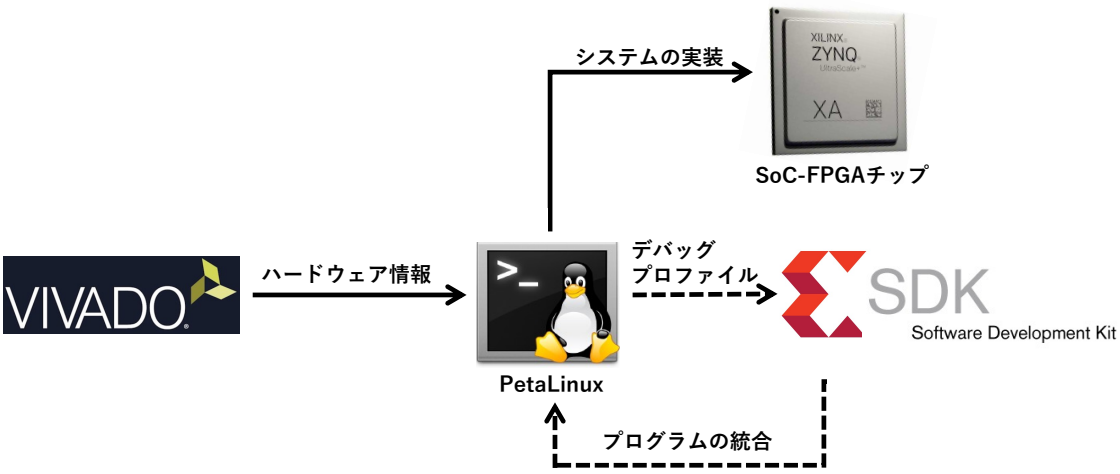


図 2.4: 機能設計の流れ

2.3.1 Vivado での開発

Vivado では論理回路設計や PL/PS ユニットの配線を行う。論理回路の設計をテキストベースに行う方法と、機能別に分けたコアを配置して配線を行う方法がある。それぞれ適した方法を使い分けて設計を行う。Vivado では IP (Intellectual Propaty) [8] という回路を機能単位でまとめたモジュールが無償で利用可能である。IP を利用することで効率よく開発が進められる。設計が完了したらシミュレーションが可能で、自分の設計した回路が正しく機能するか確認できる。最後に設計した信号を入出力する場所や信号のロジック/レベルを指定する。設計した回路情報を記述した bit ファイルを生成して Vivado での開発が終了する。例として、図 2.5 に LED を光らせる機能設計の様子を示す。この場合テキストデザインの方が適していることがわかる。

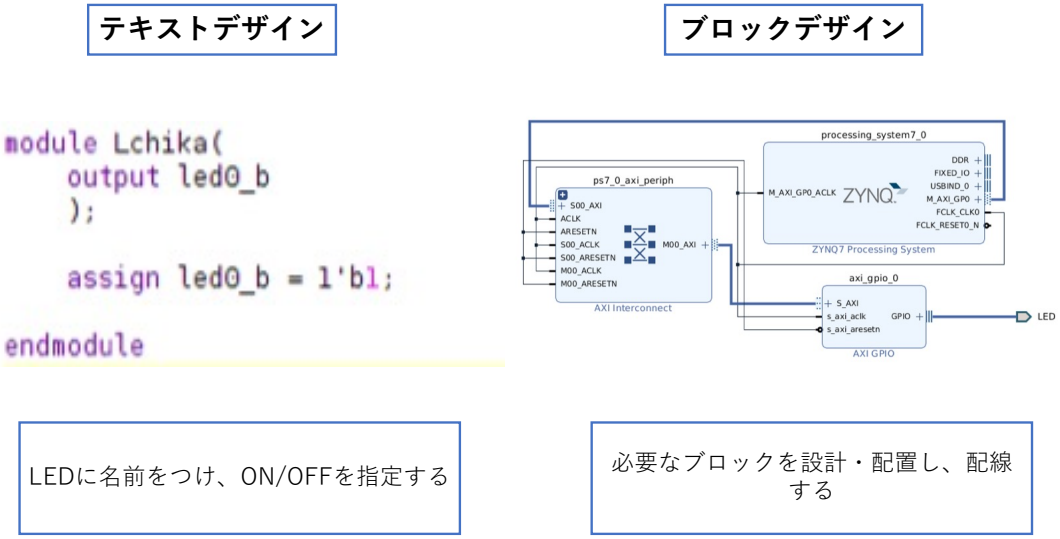


図 2.5: テキストデザイン・ブロックデザインの様子

2.3.2 SDK での開発

続いて SDK にて PS ユニットでの機能設計を行う。作成が完了した bit ファイルを引き継ぎ、これに C/C++言語でプログラムを書き加えることができる。プログラムが完成しコンパイルを行うと、自動的に PetaLinux で使う形式のファイルが作成される。PetaLinux を使用しない場合、ここで設計した機能を開発ボードにプログラムすることが可能である。UART などのシリアル通信を用いれば PC と開発ボードの通信が可能になる。なお、SDK での開発は特に必要とするプログラムがない場合は省略することができる。

2.3.3 PetaLinux

これまでの設計をもとに、Zynq-7000 の CPU で Linux を起動する。Linux の起動には必要な多数のファイルをその都度作成し、ブートファイルやカーネルを記述する Yocto [9] という手順が必要である。PetaLinux は Xilinx 社が開発・提供している開発ソフトウェアである。Xilinx 社は PS ユニット上で Linux 環境をカスタマイズ／ビルドするために必要なものを全て提供している。シリアル通信ができるコンソールを用いて外部制御が可能であり、SDK で設計した CPU 機能やプログラムを実装できる。これでシステム全体を通しての開発が終了する。

第3章　回路設計

3.1　制御信号生成回路

シリコンセンサのデータ信号を外部に出力するには制御信号が必要である。外部からの信号でセンサの動作開始や、各チャンネルのデータを出力するタイミングを決める必要がある。そこでFPGA を用いて 3 種類の制御信号並びにその生成回路を設計した。

必要な制御信号は

- クロック信号。この信号を基準としてセンサや他の制御信号が動作する。
- リセット信号。センサのデータ出力開始タイミングを決定する。
- ゲイン信号。センサの出力信号の振幅を 2 倍に増幅することができる。

の 3 種類である。生成した信号はセンサへと出力され、制御信号として機能する。設計した具体的な制御信号を以下に記す。

3.1.1　クロック信号

クロック信号は一定の時間間隔で High と Low の状態を繰り返すパルス信号である。多くの信号の基準となるとても重要な信号である。今回使用したセンサは 1 MHz のクロック信号に同期して動作するように設計されている。開発ボード Eclypse Z7 は固定 125 MHz のクロック信号を出力する発振回路をもつ。これを基準に目的の周波数のクロック信号の設計を行う。

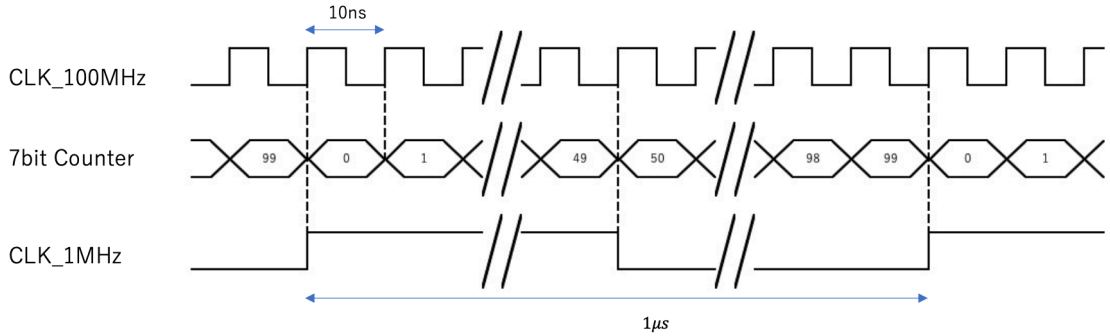


図 3.1: 1 MHz クロック生成の様子

図 3.1 は 1 MHz のクロック信号の生成の仕組みである。周波数の制限があり、IP を用いて直接 1 MHz のクロック信号を生成することはできない。まずは IP で 100 MHz のクロック信号を生成した。これを基準に 100 倍の周期を持つパルス信号を設計した。1 bit レジスタと 7 bit のカウンタを導入する。100 MHz のクロックの立ち上がりに合わせて 0 からカウントをさせる。カウントが 49 の時にレジスタを立ち下げ、99 の時に立ち上げ、カウンタを 0 に戻す。これを繰り返すこと

で 1 bit レジスタは 1 MHz のクロックとして機能する。図 3.2 はこのシミュレーション結果である。上から 125 MHz, 100 MHz, 1 MHz のクロック信号である。

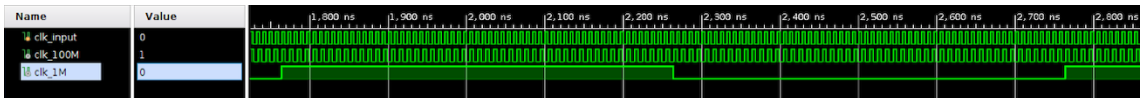


図 3.2: クロック信号のシミュレーション結果

3.1.2 リセット信号

リセット信号はセンサのデータ信号出力タイミングを決定する信号として機能する。リセット信号の立ち下がりに同期してセンサは出力を開始する。

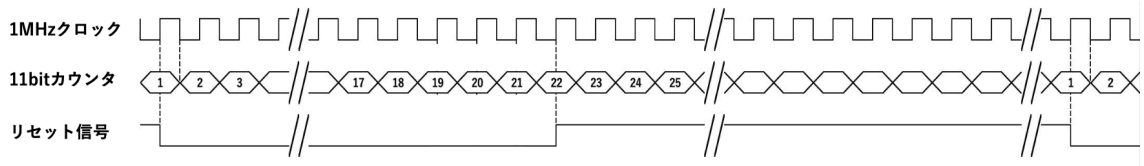


図 3.3: リセット信号生成の様子

図 3.3 はリセット信号の生成の仕組みである。リセット信号は 1 MHz クロック信号と同期する必要がある。よって 1 MHz クロックを 11 bit カウンタを用いてカウントする。立ち下がり時間は 20 クロック (=20 μs) 以上と指定されているので、21 クロック分立ち下げる。その後センサが全 128 チャンネルのデータを出力し終えた後に再び立ち下げる。以降これを繰り返す。ここでリセット信号が立ち上がっている時間はセンサの電荷蓄積時間と同等であり、この時間が長いほど各チャンネルの出力は大きくなる。図 3.4 はリセット信号のシミュレーション結果である。

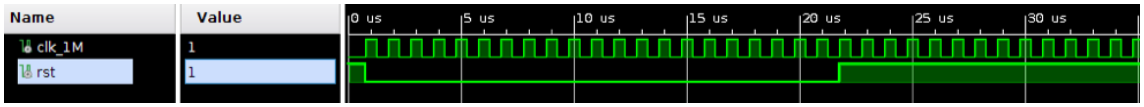


図 3.4: リセット信号のシミュレーション結果

3.1.3 ゲイン信号

ゲイン信号はクロック信号から独立した 1 bit の非同期信号である。ゲイン信号が High の時にセンサ出力信号の振幅は 2 倍になり、Low の時はそのままである。今回はゲイン信号は常に High の状態で設計した。

3.1.4 差動信号

デジタル信号は High と Low、もしくは 0 と 1 のふたつの状態で表される。その信号伝達方法には、シングルエンド方式と差動伝達方式が存在する。シングルエンド方式では基準電圧であるグランド (GND) との差で High/Low の状態を判別する。よって必要な信号線は 1 本である。このような信号をシングルエンド信号という。対して差動伝達方式では、正極と負極の 2 本の信号線

を用いる。信号線の電圧差で High/Low の状態を判別する。これを差動信号という。図 3.5 に示すように、シングルエンド信号はノイズ耐性が低く、ノイズが入ると信号が乱れてしまう。対して差動信号では対称な 2 本の信号線の電圧差で判別するためノイズがキャンセルされ、ノイズ耐性が高い。このため高速通信などには差動信号が適している [10]。今回は 3 つの制御信号に加えてそれらを差動信号化する回路を設計した。

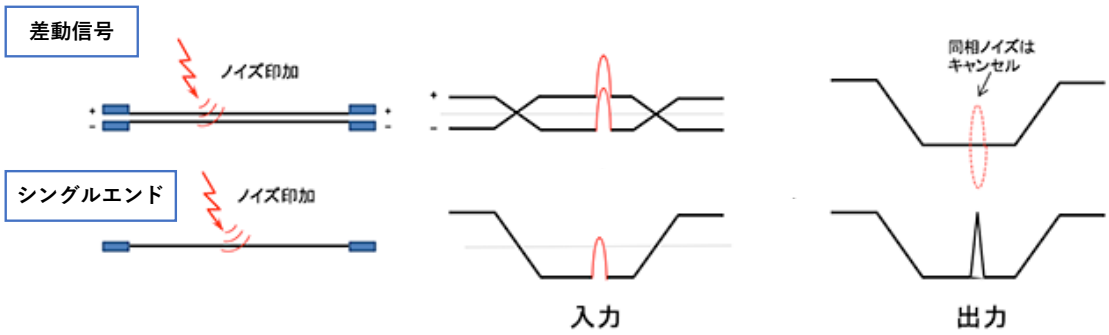


図 3.5: シングルエンドと差動信号 [11]

3.2 A/D 変換回路

センサの出力信号をデジタル信号に変換する回路である。

- Pmod AD1 と開発ボードの通信確立
- Pmod AD1 に正しい制御信号を送る

この 2 つの機能を持った回路を設計する必要がある。

3.2.1 A/D 変換の仕組み

12 bit の解像度をもつ ADC が行っている A/D 変換の流れを以下に示す。入力されるアナログ信号を離散的なデジタル信号に変換する行程を量子化という。変換するために入力信号を 2^{12} 段階に分ける必要がある。よってアナログ入力の電圧範囲が定まっていなければいけない。電圧の最大値は V_{DD} で表される。入力電圧範囲は 0 〜 V_{DD} である。Pmod AD1 の規格は $V_{DD} = 2.35$ 〜 5.2 V である。12 bit の場合、000000000000〜111111111111 (0 〜 4095) の 4096 段階 (図 3.6 の縦軸) に変換している。1 段階の幅は 1 LSB (Least Significant Bit) と表される。使用した ADC では

$$1\ LSB = \frac{V_{DD}}{4096} \tag{3.1}$$

である。例を挙げると $V_{DD} = 5.0$ V のとき、式 3.1 より 1 LSB ≒ 0.0012 なので 0 〜 0.0012 V までは 000000000000 で表される。以上の様にして 12 bit のデジタル信号が出来上がる。図 3.6 に A/D 変換の対応を示す。

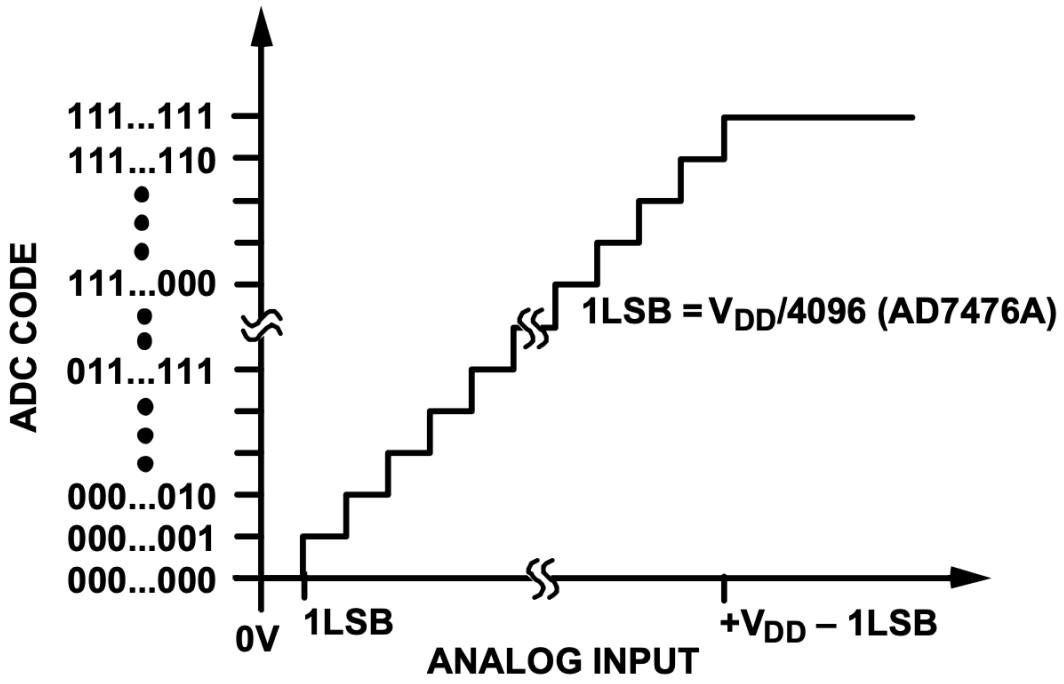


図 3.6: A/D 変換の様子 [13]

3.2.2 ADC の接続

図 3.7 に Pmod AD1 を接続したボードの様子を示す。しかし、回路を繋がなければ ADC は機能しない。ADC と開発ボード間の通信インターフェースの確立が必要である。Vivado の IP を用いて Pmod AD1 とボードの Pmod ポートを通して Zynq-7000 の I/O ポートに接続し、PS ユニットで制御する設計になっている。

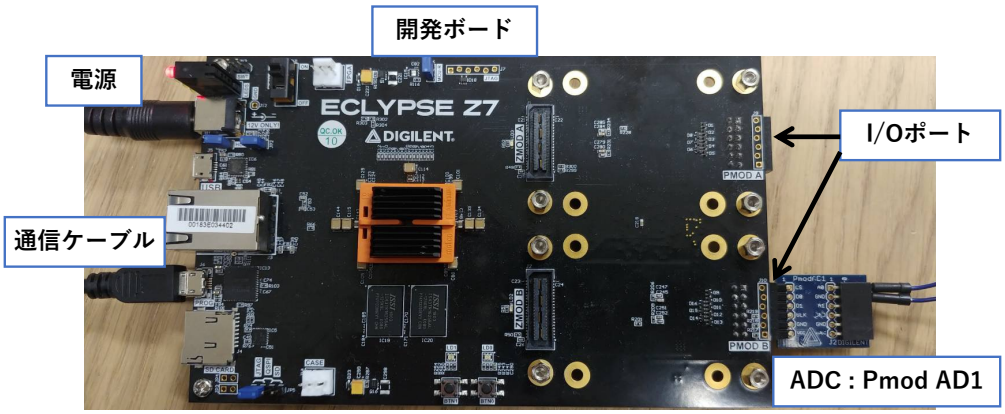


図 3.7: ADC 接続時の外観

実装する通信インターフェースは図 3.8 に示すように ADC を制御するための PS ブロック (左) と、ADC を接続するためのブロック (右) からなる。これらをつなげる回路を設計する。

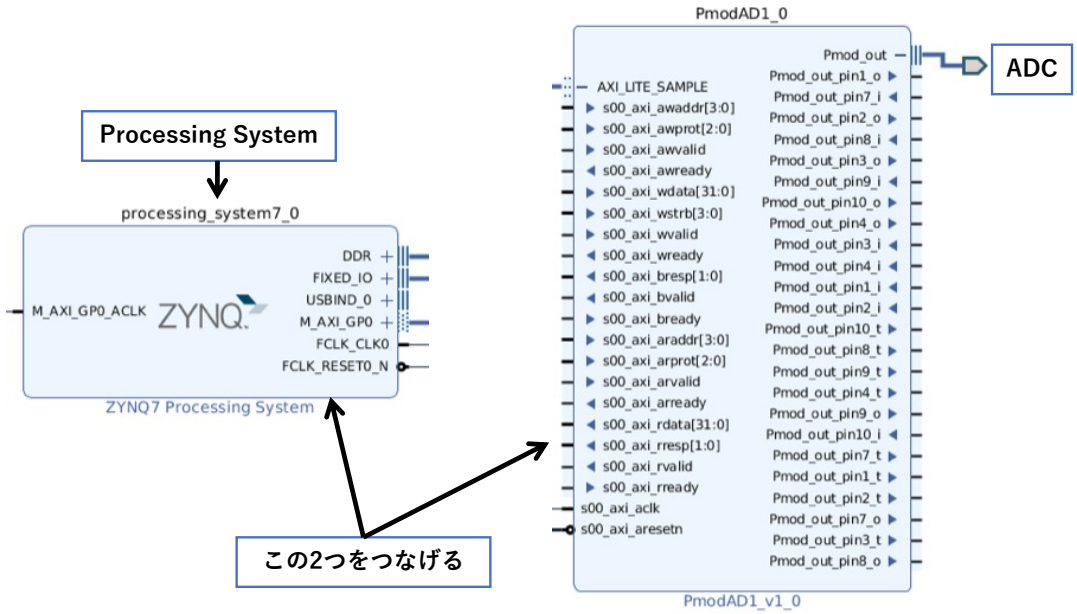


図 3.8: インターフェース確立のためのブロック

3.2.3 データ取得タイミング

使用した Pmod AD1 は SPI (Serial Peripheral Interface) という通信プロトコルを使用してボードと通信している。動作にはマスターであるボードからのクロック信号が必要である。マスター

クロックには 100 MHz クロック信号を利用した。CS (Chip Select) 信号はマスタークロックを認識すると自動で駆動する。CS 信号に同期して 12 bit データをボードに渡す仕組みである。データを正しく認識するために、初めの 4 クロックは 0 を出力し続ける。よって 1 信号のデータサイズは 16 bit になる。A/D 変換されたデータは PL ユニットに入力され、AXI を経由して PS ユニットへ渡される。ADC のタイミングチャートを 12 bit で 110100110001 (=3376 ADC) の値に変換された場合を例に図 3.9 に示す。

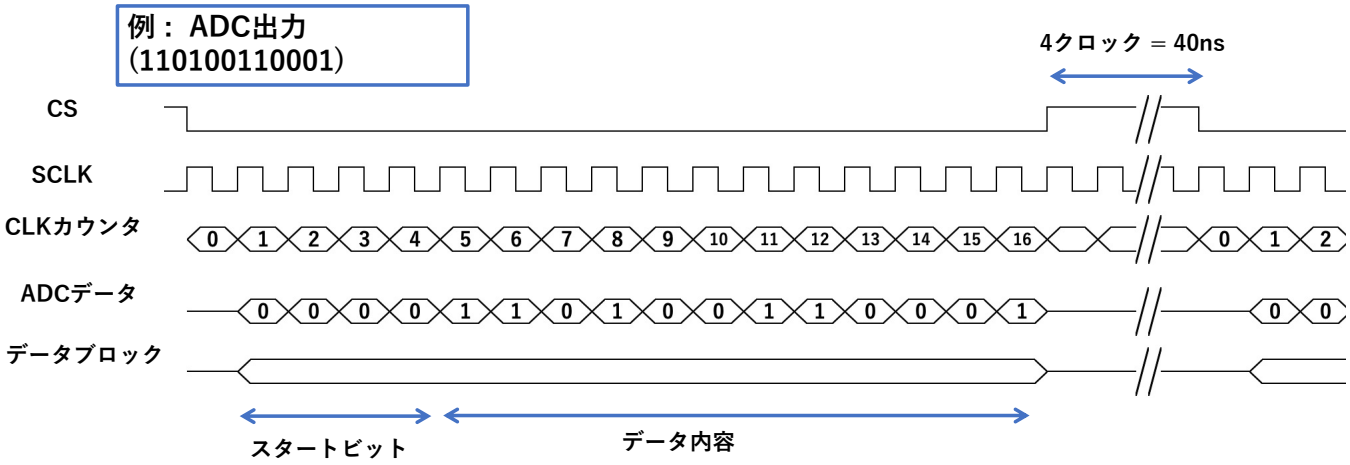


図 3.9: ADC のタイミングチャート

第4章　開発結果

4.1　達成項目

今回の取り組みでは

- 制御信号生成回路
- A/D 変換回路

の開発を完了した。以下では、実装した各回路の結果を示す。

4.2　制御信号生成回路

センサを制御するための信号を生成する回路である。クロック信号・リセット信号・ゲイン信号の 3 種類と、これらを差動化する回路を設計した。図 4.1 は制御信号生成回路の回路図である。(3.1 ”制御信号生成回路” を参照)

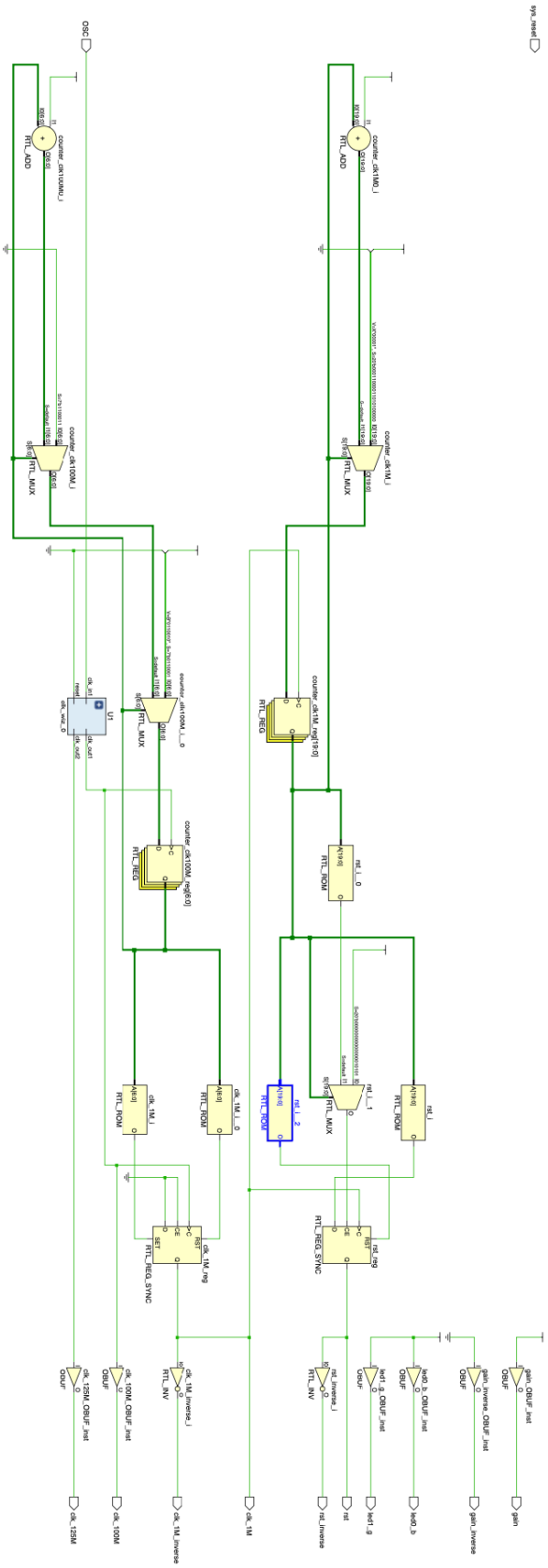


図 4.1: 制御信号生成の回路図

4.2.1 クロック信号

以下の図は 125 MHz・100 MHz・1 MHz のクロック信号の波形をオシロスコープで観測したものである。それぞれ周期が 8 ns・10 ns・1 μs であることが確認できる。125 MHz・100 MHz についてはオシロスコープやプローブの帯域幅 [12] により矩形波として観測はできないがシステム上では 8 ns・10 ns の周期で High/Low を繰り返すデジタル信号として処理されている。

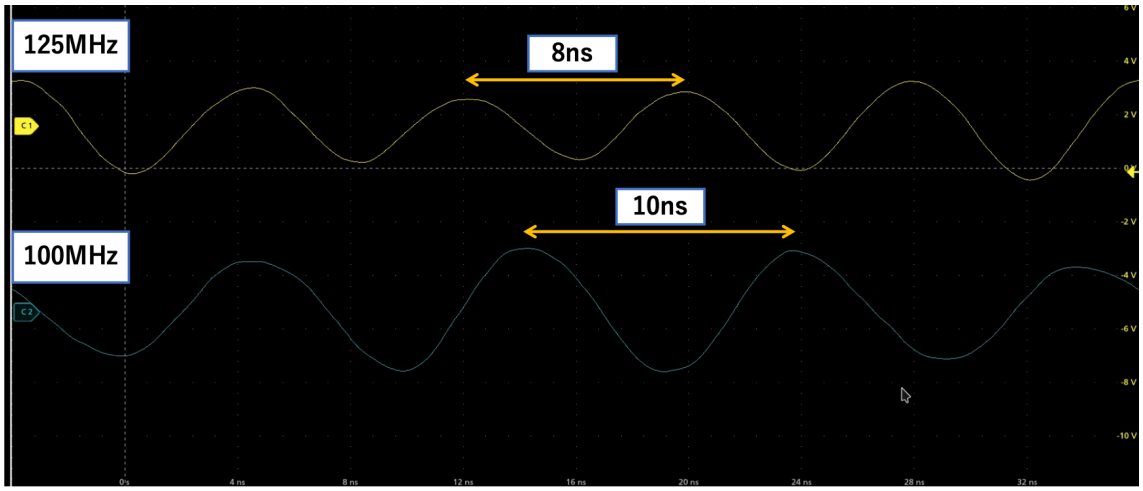


図 4.2: クロック信号の比較 (125 MHz・100 MHz)

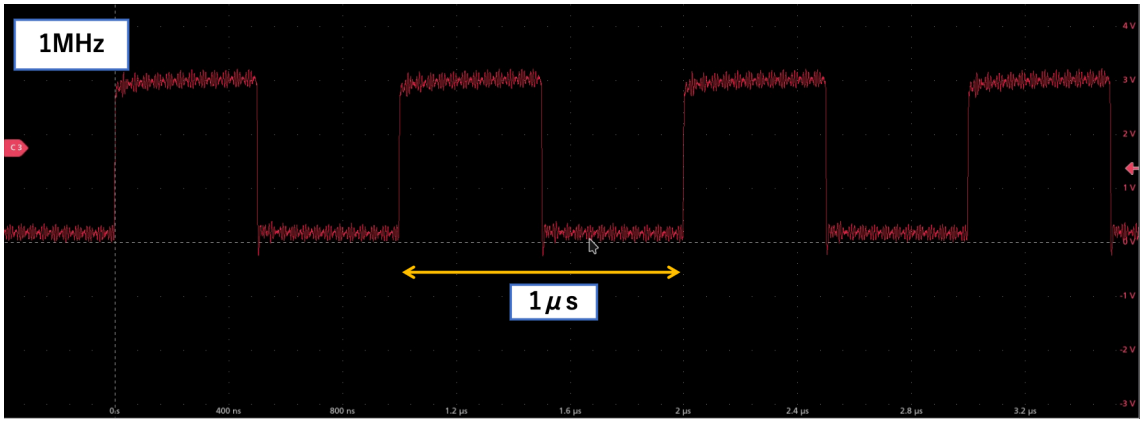


図 4.3: 1 MHz クロック信号

4.2.2 リセット信号

図 4.4 はリセット信号と 1 MHz のクロック信号の実際の波形である。リセット信号はクロック信号の立ち上がりのタイミングで立ち下がっており、21 クロック ($= 21$ μs) 後に再び立ち上がっている様子が確認できる。また、図 4.5 に示すようにリセット信号の周期は 1 ms であり、センサのデータ蓄積時間は 1 ms になっている。プログラムを書き換えることで蓄積時間は変更可能である。

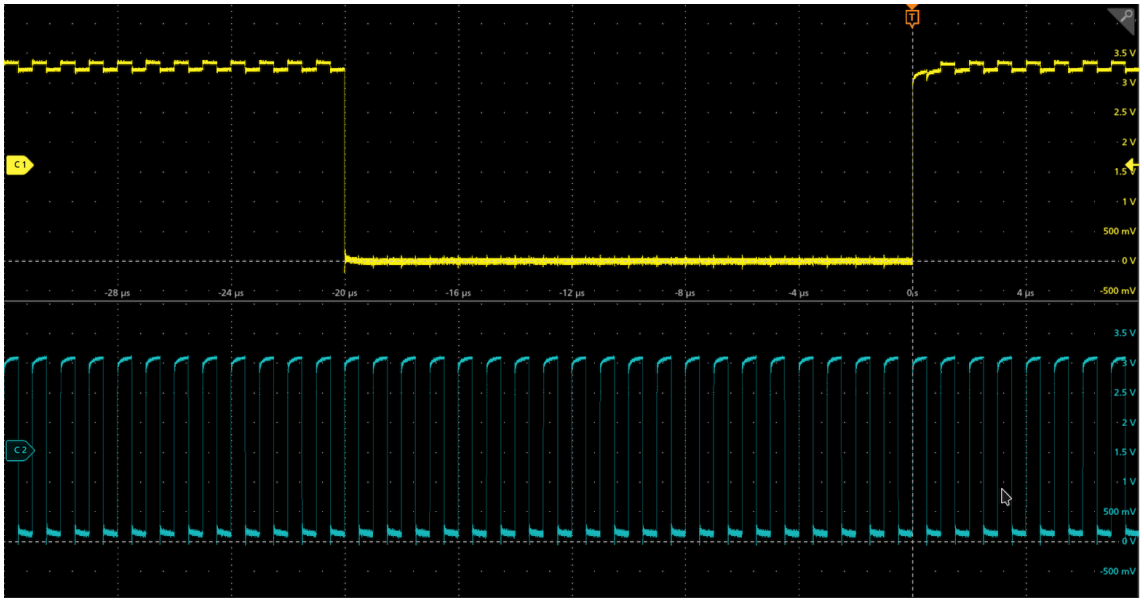


図 4.4: リセット信号・クロック信号の比較

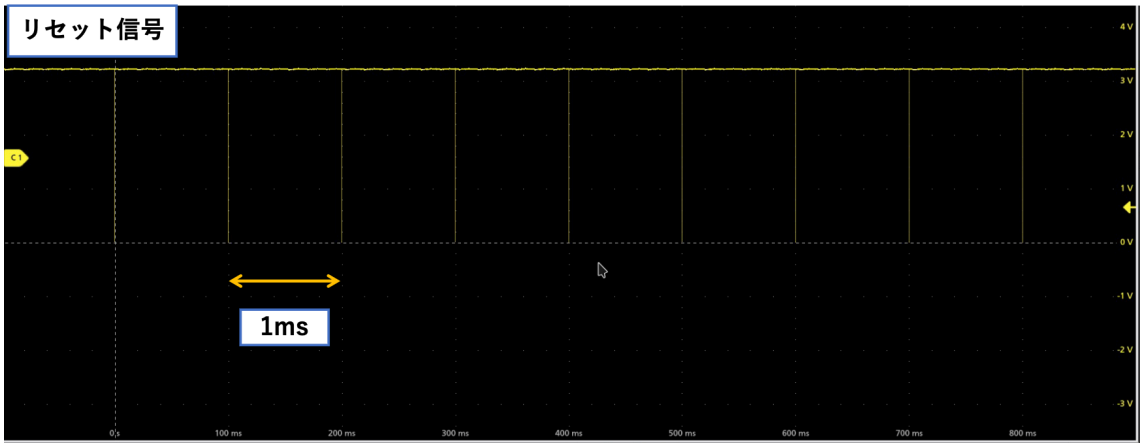


図 4.5: リセット信号の周期

4.2.3 ゲイン信号

ゲイン信号は High/Low どちらかの状態を出力し続ける非同期信号である。今回は High の状態を出力し続けるように設計した。図 4.6 は実際の波形である。こちらもプログラム書き換えによって変更可能である。開発ボードにはプッシュボタンと RGB-LED が 2 つずつ搭載されている。図 4.7 にボード機能であるボタン・LED の実際の様子を示した。2 つの LED はそれぞれ赤・青・緑に光り、LED の下にボタンがある。ボタンを押すとゲイン信号の High/Low が切り替わり、インジケーターとして LED を利用する設計を考えている。

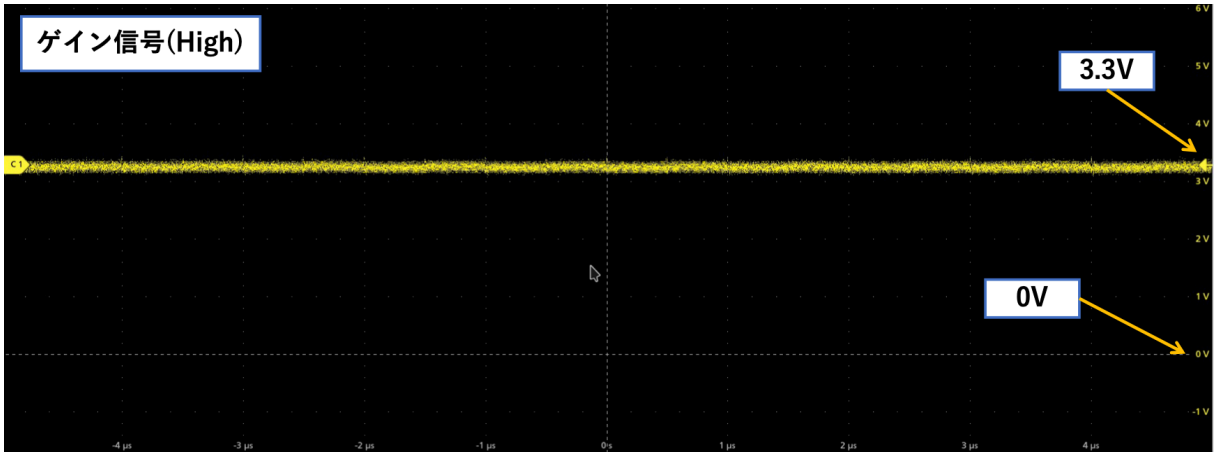


図 4.6: ゲイン信号 (High)

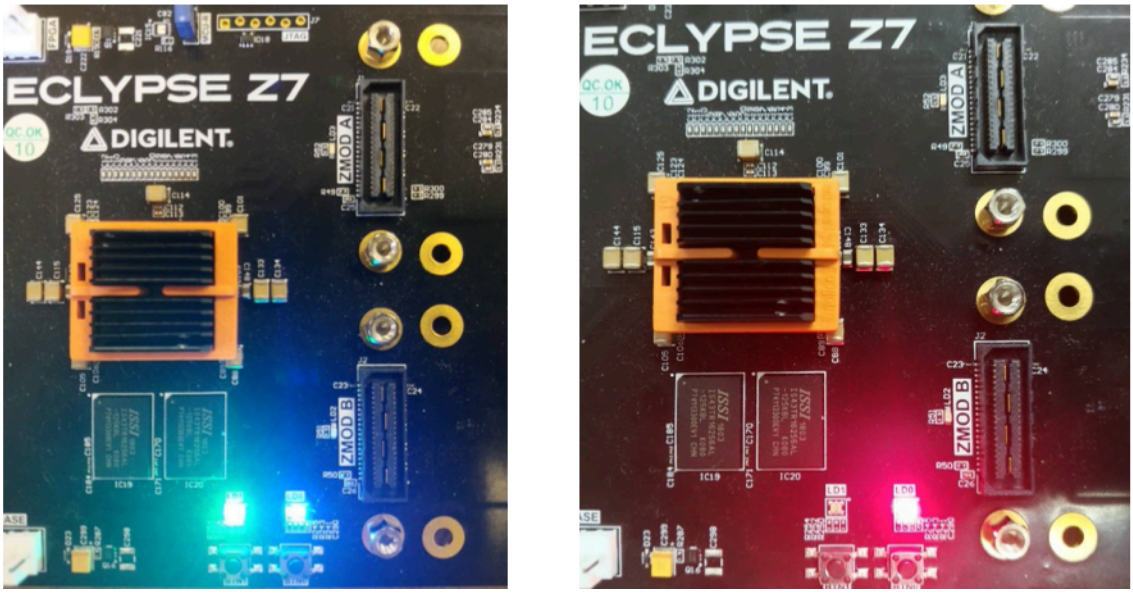


図 4.7: ボタン・RGB-LED

4.3 A/D 変換回路

4.3.1 データ信号の流れ

センサからのアナログデータ信号をデジタル変換する回路である。図 4.8 に設計した回路全体のブロック図を示す。

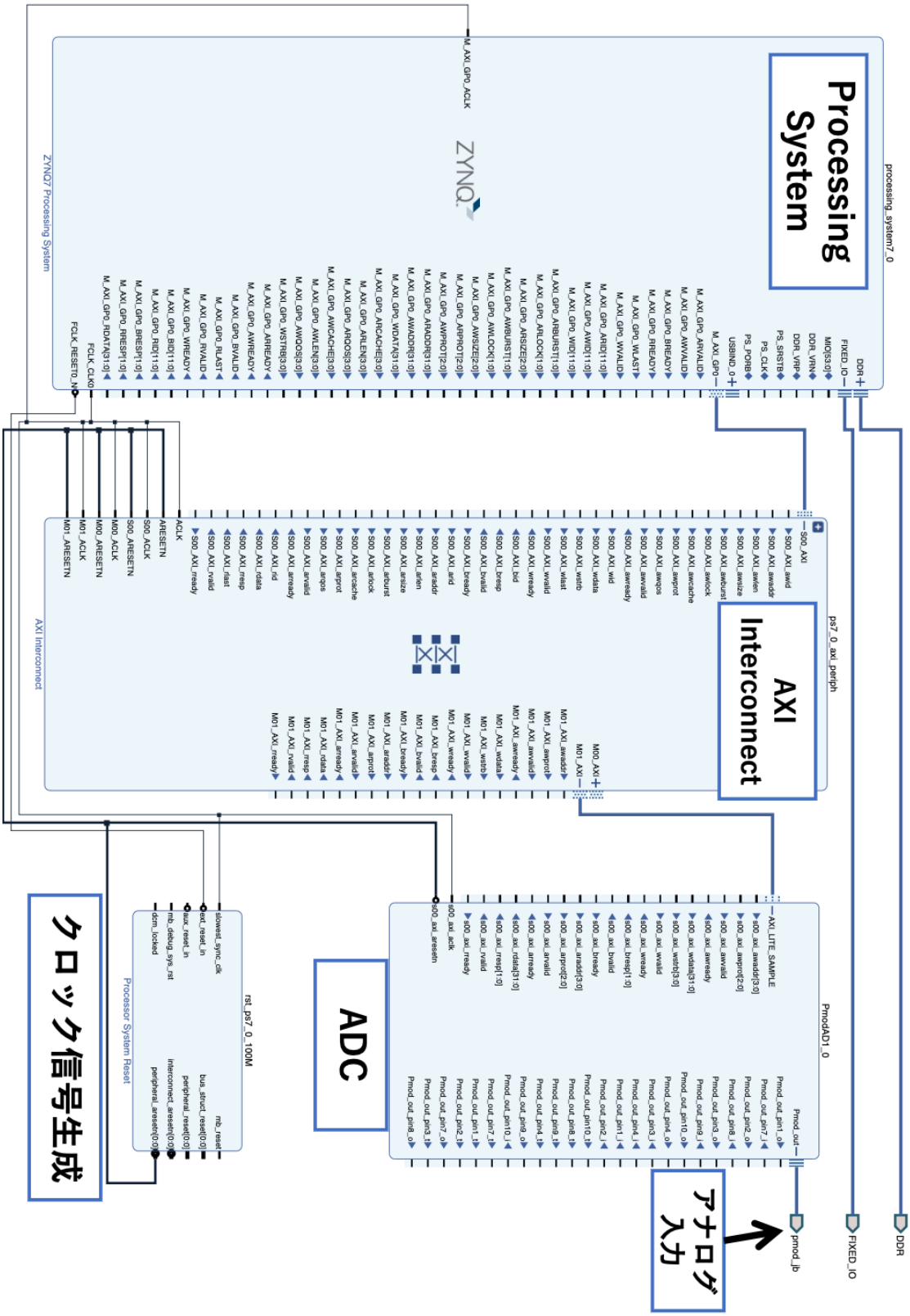


図 4.8: A/D 変換回路のブロック図 (全体図)

各ブロックをつなげる太い青線をインターフェースといい、複数の信号で通信している。同一ブロック内部のマスターポートに入力された信号はそのままスレーブポートに渡され、出力される。マスター・スレーブとは優先順位を表す仕組みである。図 4.8 は全体のブロック図である。これより ADC (右) と PS ユニット (左) は AXI (中央) を通して繋がっていることがわかる。マスタークロックはクロック生成ブロック (右下) の slowest_sync_clk から出力され、各ブロックの緑色で示した部分に入力されている。これでこれらのブロックはマスタークロックに同期して動作が可能になる。以下に ADC の出力であるデジタルデータ信号の流れを示す。

- ADC ブロックの pmod_out_pin2_i と pmod_out_pin3_i に入力される
- ADC ブロックの AXI_LITE_SAMPLE インターフェースの s00_axi_rdata[31:0] から出力される
- AXI ブロックの M01_AXI インターフェースの M01_AXI_rdata に入力される
- AXI ブロックの S00_AXI インターフェースの S00_AXI_rdata から出力される
- PS ブロックの M_AXI_GP0 インターフェース内部の M_AXI_GP0_RDATA[31:0] に入力される

以上のような経路をたどって ADC のデジタル出力信号は PS ユニットまで運ばれ、様々な処理が可能になる。また、図 4.9 は図 4.8 の PS 周りのブロックを、図 4.10 は図 4.8 の ADC 周りのブロックを、図 4.10 は実際の ADC の接続の様子を表している。

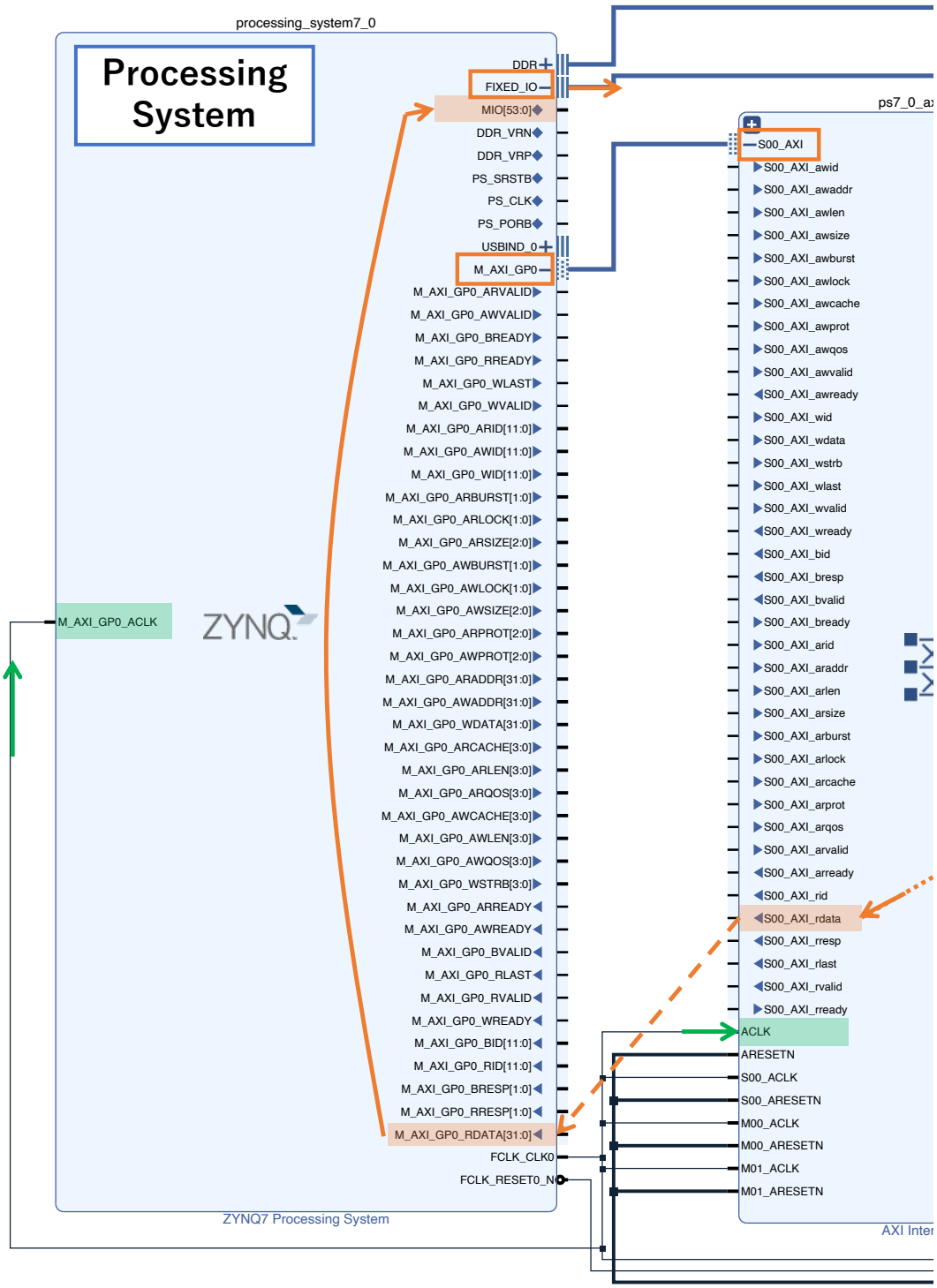


図 4.9: A/D 変換回路のブロック図 (PS)

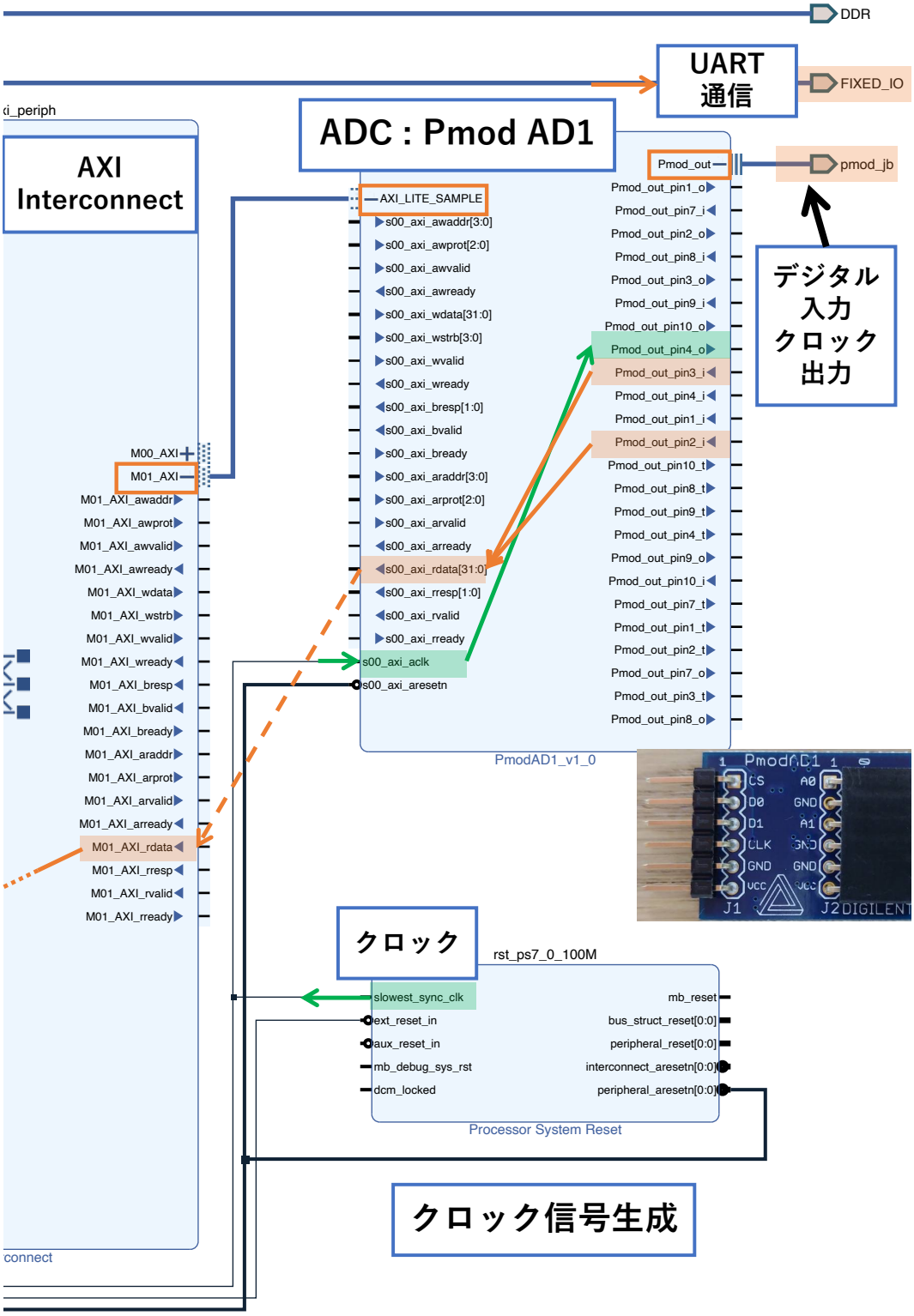
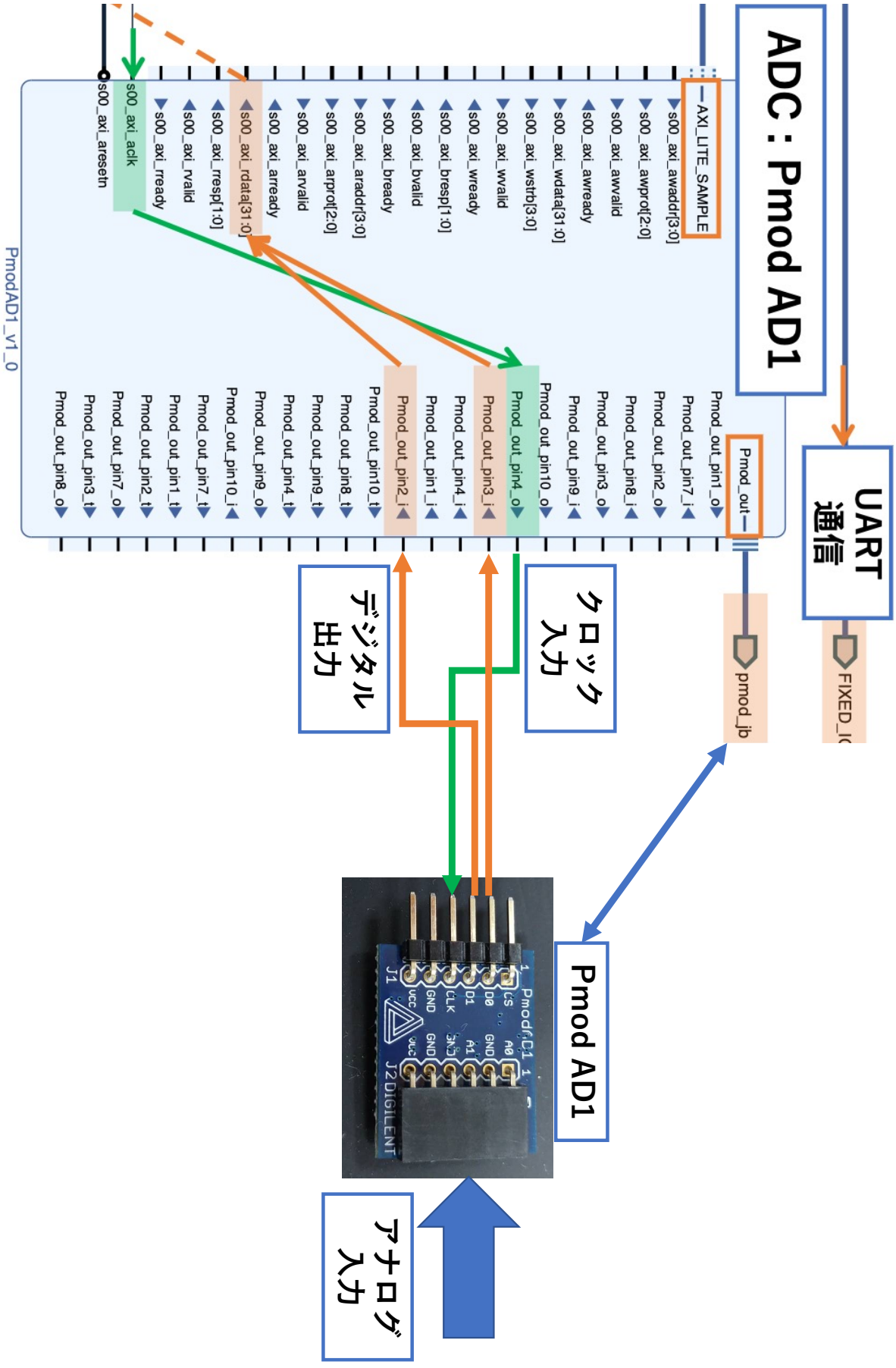


図 4.10: A/D 変換回路のブロック図 (ADC, クロック)


ADC : Pmod AD1
UART
通信
FIXED_I
pmod_jb
PmodAD1_v1_0
AXI_LITE_SAMPLE
s00_axi_awaddr[3:0]
s00_axi_awprot[2:0]
s00_axi_awvalid
s00_axi_awready
s00_axi_wdata[31:0]
s00_axi_wstrb[3:0]
s00_axi_wvalid
s00_axi_wready
s00_axi_bresp[1:0]
s00_axi_bvalid
s00_axi_bready
s00_axi_araddr[3:0]
s00_axi_arprot[2:0]
s00_axi_arvalid
s00_axi_arready
s00_axi_rdata[31:0]
s00_axi_rresp[1:0]
s00_axi_rvalid
s00_axi_rready
s00_axi_aclk
s00_axi_aresetn
Pmod_out
Pmod_out_pin1_o
Pmod_out_pin7_i
Pmod_out_pin2_o
Pmod_out_pin8_i
Pmod_out_pin3_o
Pmod_out_pin9_i
Pmod_out_pin10_o
Pmod_out_pin4_o
Pmod_out_pin3_i
Pmod_out_pin2_i
Pmod_out_pin1_i
Pmod_out_pin4_i
Pmod_out_pin10_t
Pmod_out_pin8_t
Pmod_out_pin9_t
Pmod_out_pin4_t
Pmod_out_pin9_o
Pmod_out_pin10_i
Pmod_out_pin7_t
Pmod_out_pin1_t
Pmod_out_pin2_t
Pmod_out_pin7_o
Pmod_out_pin3_t
Pmod_out_pin8_o
クロック
入力
デジタル
出力
Pmod AD1
アナログ
入力
J1
VCC
GND
CLK
D1
D0
CS
J2 DIGILENT
GND
3V3
A1
A0
GND
VCC

4.3.2 通信テスト

図 4.9 の PS ブロックの M_AXI_GP0_RDATA[31:0] に格納されているデータ信号を、 FIXED_IO インターフェースに渡して図 4.10 の FIXED_IO を通して外部 PC と通信を行う。

図 4.9 の FIXED_IO インターフェース内部の MIO (Multiplexed I/O) の中には各種通信に必要な機能があらかじめ設計されている。この中の外部機器とシリアル通信をするための最も単純な UART という通信を用いて A/D 変換回路の動作確認を行った。

アナログ入力には一定の電圧出力を持つ乾電池を用いた。分圧抵抗回路を挟むことで乾電池の電圧を下げ、複数の電圧入力に対しての ADC の出力データをとった。分圧抵抗回路の抵抗では、抵抗 R_1,R_2 [Ω] を変更することで乾電池の電圧 V_{CC} に対して出力電圧 V_{Out}

$$V_{Out} = \frac{R_2}{R_1 + R_2} \times V_{CC} \tag{4.1}$$

が得られる。
図 4.12 に実際に使用した分圧抵抗回路とその回路図を示す。

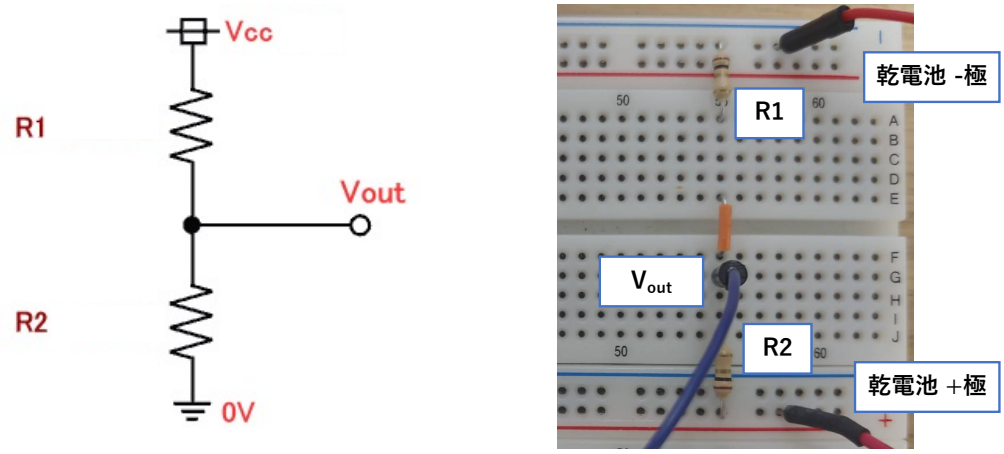


図 4.12: 分圧抵抗回路 [14]

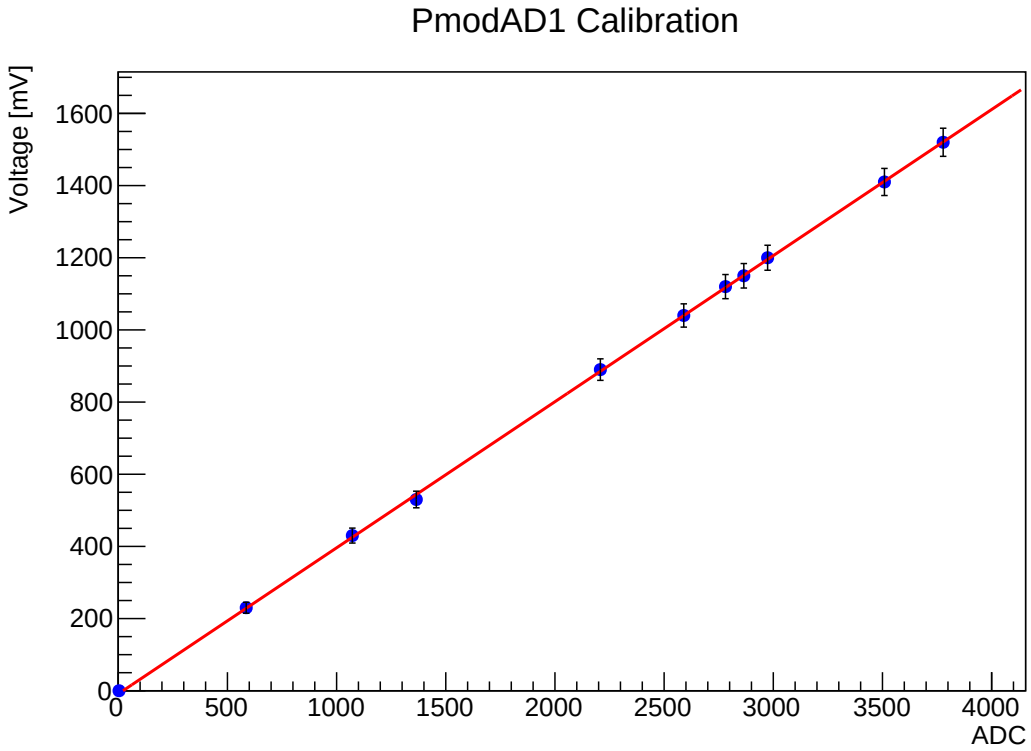


図 4.13: ADC 校正曲線

図 4.13 は入力した電圧 (mV) を縦軸に、ADC の出力を縦軸に線形近似を行った結果である。

$$Voltage[mV] = (0.405 \pm\ 0.008)ADC - (8.76 \pm\ 15.8) \tag{4.2}$$

式 4.2 の校正式を得られた。このことからこの ADC の分解能は約 0.405 mV であること、そして A/D 変換回路が正しく動作していることが確認できた。

第5章　まとめ

5.1　開発成果

読出回路構築に向けて必要な 4 つの機能

- 制御信号生成
- A/D 変換
- データ処理
- データ伝送

のうち、制御信号生成と A/D 変換の機能をもつ回路の設計・実装を完了した。

5.2　将来展望

今回開発を目指したデータ処理・データ伝送回路はいずれも CPU 機能を利用したものである。データ処理はデータのない部分を圧縮するゼロサプレッション処理や、連続したデータがある部分をまとめる処理などの圧縮処理をリアルタイムに行うことができる。データ伝送回路では GbEthernet を通して最大 1Gbps のデータ通信が可能である。これらの回路の開発を行うことで、用意できる環境で最も早い速度でセンサ信号読出が行えると考えている。引き続きこれらの回路の実装に取り組んでいきたい。

謝辞

本研究では、たくさんの方に大変お世話になりました。指導教員である山口頼人准教授は、問題点が出た時に親身になって原因究明に付き合ってくださったり、理解の助けになる助言を多数くださいました。また専門的な知識や技術が必要になる場面では、長崎総合科学大学の大山健教授、浜垣秀樹特命教授、そしてサイエナジー社の阿部圭一様には大変お世話になりました。皆様の助力には非常に感謝しています。志垣賢太教授には、研究に向き合う熱心な姿勢をその助言や立ち居振る舞いから教えていただきました。忘年会では自宅に招いていただいたりと非常に親身になって接してくださいました。本間謙介准教授は、いつも斬新な切口からの質問をいただき、自分の見識を広げてくださいました。また、ラボエクササイズでの実験の作法を教えていただきました。三好隆博助教には、発表があるたびに的確なアドバイスや質問をいただいたり、研究に必要な PC の準備やラボエクササイズでお世話になりました。八野哲助教にはミーティングの時に積極的に質問をいただき、励みになりました。また、ラボエクササイズでデータ解析手法を教えていただきました。荻野雅紀さんは専門知識が必要な場面ではいつも力を貸していただきました。そして、研究室の同期・先輩方にも大変お世話になりました。山川さんは、自分に東プレ REALFORCE を譲っていただきました。大佐古さんは時々遊びに来てくださったり飲みの席に来てくれたりと、貴重な話を聞かせてくれました。大矢さんは、自分が大きな失態を犯してしまった時に助けていただきました。桐田さんは、資料作成のポイントを教えてくださいました。キムさんは、同室の先輩としてたくさんのことを教えていただきました。徳本さんはキムさんと共に飲みに連れて行ってくれました。石橋さんは僕にベジという名前をくださいました。江島さんには PC 環境構築についてたくさん教えてもらいました。重國さんは自分の経験を伝えてくださり、大学院試験などで助けてもらいました。同期の 4 年生のおかげで賑やかに過ごせました。チンチラのうんちゃんは、自宅で遊んでくれて寂しさを忘れさせてくれました。最後にいつも応援してくれていた家族と友人たちに感謝します。

参考文献

[1] ”Particles and nuclei”(2019) by Bogdam Povh, Klaus Rith, Christoph Scholz, Frank Zetsche,
transrated by Toshiaki Shibata, 丸善, p.354

[2] 永江知文, 永宮正治 共著 (2011)「原子核物理学」, 裳華房, p.91

[3] https://www.portwell.co.jp/blog/difference_between_fpga_and_cpu Portwell, なぜ今”FPGA”が注目されているのか。

[4] https://www.iri-tokyo.jp/uploaded/attachment/2471.pdf 組み込みシステムは FPGA/SoC の時代へ

[5] https://www.rohm.co.jp/electronics-basics/ad-converters/ad_what2 ローム　A/D コンバータとは?　基本動作

[6] https://japan.xilinx.com/products/silicon-devices/soc.html Xilinx SoC

[7] https://japan.xilinx.com/products/silicon-devices/soc/zynq-7000.html Xilinx ZYNQ-7000 SoC 製品の特徴

[8] https://japan.xilinx.com/products/intellectual-property.html Xilinx IP

[9] https://qiita.com/AngryMane/items/61d2fa47246a9f9217f5 Qiita Yocto の概要

[10] ノイズ対策ドットコム, 差動信号

[11] https://edn.itmedia.co.jp/edn/articles/1706/28/news014_3.html　EDN Japan 作動信号伝送のメリット

[12] https://www.gec-tokyo.co.jp/design-notes/no-002_oscilloscope-measuring-techniques-part-2 グローバル電子株式会社　オシロスコープの測定技術

[13] ANALOG DEVICES, AD7476A Datasheet, p.15

[14] https://www.kairo-nyumon.com/resistor_divider.html 電子回路設計入門サイト 分圧抵抗