ReVolver/C40 の
シェーディングステージの設計と実装

吉谷 直樹

平成 9 年 2 月 14 日

Abstract

3 次元空間に存在する物体を、単位立方体 (ボクセルと呼ぶ) の集まりで構成されていると考え、この空間をボリュームデータのまま 2 次元のスクリーンに投影することによって、情報の可視化を行う技法をボリュームレンダリングと呼ぶ。ボリュームレンダリングでは、内部情報を持つデータの可視化が行えることから、物体の表面だけでなく、物体の内部や背後の状況も画像として表現することができる。そのため、医療分野における人体データの可視化、構造解析や流体計算をはじめとする科学技術計算の結果の可視化などの手段として重要視されている。しかし、この手法の実現には膨大な記憶容量と計算量を要する。そこで当研究室では、ボリュームレンダリングを高速に行う専用並列計算機 *ReVolver*/C40 を開発中である。*ReVolver*/C40 は画像生成処理を視線生成/ピクセル値計算/シェーディング処理という 3 つの処理に分割し、パイプライン処理を行うことによって高速表示を可能としている。上記の 3 つの処理をするステージをそれぞれ、視線生成ステージ (Ray Casting Stage:RCS)、ピクセル値計算ステージ (Pixel Calculation Stage:PCS) 、シェーディングステージ (Shading Stage:SS) と呼ぶ。本稿では、*ReVolver*/C40 のシェーディングステージの構成、設計および実装について述べる。

シェーディング処理とは、光源情報を用いて生成画像の輝度を補正することにより、画像に現実感を持たせる処理であり、*ReVolver*/C40 では depth gradient shading 方式を採用している。この手法は、スクリーン上の各ピクセルごとの視点と表面ボクセル間の距離 (深さ) を用いて表面の勾配を計算し、ピクセルの輝度を求めるピクセルアプローチの一種である。また、勾配計算の際に X/Y 方向それぞれについて目的のピクセルを中央に含む数ピクセルの区間領域を取り出して、その両端の深さを用いて勾配を求めているため、処理内容が簡素で高速化に適している。このため、種々のボリュームレンダリング専用計算機に採用されている。

シェーディングステージでは、depth gradient shading の手法を用いて、ピクセル値にシェーディング処理を行うが、その際に必要なデータには局所性がある。よって、ここでは画面を 4×4 の領域に 16 分割し、それぞれを一つの Processing Element(PE) に割り当てることで並列処理を行い高速化を図っている。一つの PE は、ピクセルデータを受け取るための Receive Buffer(RB) と画像出力のための Output Buffer(OB) を持ち、中心にシェーディングを行う Digital Signal Processor (DSP) を搭載している。16 台の PE を総称して、Shading Unit(SU) と呼ぶ。ピクセル値計算ステージからの入力、及びディスプレイへの出力は、逐次的である。しかし、SU で並列処理を行うために、PCS から逐次的に送られてくるピクセル値の各 PE への分散、各 PE で処理した画像データの収集という煩雑な処理が必要となる。本 SS では、これらの煩雑な処理を PE から切り放し、それぞれ独立のハードウェアで行うことで SU の高速化を図っている。ピクセル値を受け取り 16 個の PE に分配する Pixel Transceiver(PT) と、画像出力を行う Video Signal Generator(VSG) がこれらの処理の要となる。

Pixel Transceiver は、各ピクセルのデータをピクセル値計算ステージから受け取り、そのピクセル座標をもとに、担当の PE とメモリアドレスを計算し、そのアドレスにデータを書き込む。PE は

depth gradient shading 方式で自分の担当領域のシェーディング処理を行うために、自分の領域にない depth 情報を用いて法線を計算する必要がある。PT がこの depth 情報を各 PE に重複して持たせる方法をとると、計算時間がかかりパイプラインを止めてしまう。そこで depth 値を DSP 間通信によって取得すると、PT はすべての処理を全体のパイプラインを止めることなく処理を終えることができた。

Video Signal Generator は、各 PE の OB からシェーディング処理を終えたデジタル RGB データを受け取り、アナログ信号に変換して CRT に出力する。この部分では、データ転送要求と画像データ出力のタイミングが重要である。CRT への表示は、512^2 ピクセルの表示画面サイズで 1 秒間に 60 枚の画像を表示できる設計になっている。

PT ならびに VSG の CRT Controller を FPGA(Field Programmable Gate Array:10000 ゲート相当、遅延時間 $5ns$) を用いて実装した結果、PT はゲート使用率 91%、VSG の CRT Controller はゲート使用率 77% で実現できること、ならびに十分な速度が得られることを確認した。

Abstract

Volume rendering is a technique to visualize information by projecting the three dimensional space on the two dimensional screen. The object in this space with volume data can be regarded as a group of unit cubes(They are called voxels). We can project not only the surface of objects but also the inside of and behind objects by using the volume rendering,because we can visualize the data which have internal-information. So it is an important technique to visualize the medical images of the human body and the results obtained by scientific computation such as the computation at fluid dynamics and the structure analysis. But this method requires enormous amount of storage and large computation time. So we are developing a parallel machine specially for volume rendering '*ReVolver*/C40'. This machine can display images fast by the pipelining of three independent processes. These processes are ray casting, pixel calculation and shading. Pipelining stages for these processes are called the Ray Casting Stage(RCS), the Pixel Calculation Stage(PCS) and the Shading Stage(SS), respectively. This paper describes a design of the Shading Stage of *ReVolver*/C40. The goal of the Shading Stage is to shade the images generated by the Pixel Calculation Stage.

Shading is a method to correct intensity at each pixel in a generated image with light source information, and to get the realistic image. The Shading Stage computes the intensity with the depth gradient shading method. This is one of the pixel approach,which calculates the surface gradient and the intensity of a pixel with the distance(depth) from an view point to a surface voxel at each pixel on a screen. The depth gradient shading is simple and fit for high-speed processing because it calculates a gradient with only two depths at each end pixels of a short segment including the target pixel for both x-coordinate and y-coordinate. So, this method is adopted for various machines specified for volume rendering.

When the Shading Stage does shading with the depth gradient shading method, it accesses the limited part of the screen. So, the screen is divided into $16(4 \times 4)$ components and one Processing Element is assigned to each of them. It enables parallel and high-speed processing. Each processing element has a receive buffer to receive pixel values, the Receive Buffer, and a frame buffer for image output, the Output Buffer, and has a Digital Signal Processor (DSP) as a kernel for the shading process. Shading Unit is the general term for these 16 Processing Elements. Inputs from Pixel Calculation Stage and outputs on display are successively. But it is difficult that Processing Elements compute all complicated process such as distributing pixel values received from Pixel Calculation Stage and gathering visual data and image output successively to actualize parallel processing. So it is important that they are processed by different hardware part from Processing Elements. So Pixel Transceiver, which receive pixel values and divide them among Processing Elements, and Video Signal Generator, which output images, are required.

Pixel Transceiver receives pixel values from Pixel Calculation Stage, and chooses Processor Element number and computes its memory address from pixel values, and writes pixel values to its memory. Processing Elements have to calculate vertical line with depth information, which are outside of them, to do shading with depth gradient shading method. If Pixel Transceiver distributes outside depth information to every Processor Element, PT has to calculate enormously and disturbs pipelining. Now, Processing Elements get this information by using communication port in the Digital Signal Processor,then Pixel Transceiver can finish its all processes without disturbing the pipelining.

Video Signal Generator receives digital shading - done RGB values from the Output Buffer, and transfers it to analog signal, and outputs images. It is important when Video Signal Generator requests data and outputs data. Video Signal Generator can output 60 images per a second to the 512^2 pixels display.

Pixel Transceiver and CRT Controller of Video Signal Generator are implemented in the Field Programmable Gate Array(FPGA:about 10000 gate, $5ns$ delay). As a result, Pixel Transceiver makes 91% use of gate availability, and CRT Controller of Video Signal Generator makes 77% use of gate availability, and they can calculate fast enough.

目 次

1 はじめに

現実の時空間や計算モデルにおけるオブジェクトや自然現象の観測結果の多くは、3 次元のボリュームデータで与えられる。物体内部の可視化や科学シミュレーションによって計算されたポテンシャルや力の場の可視化に直接適用することを主眼として開発された手法として、ボリュームレンダリングがある。ここで、ボリュームデータは、整数格子座標値をもつボリュームの最小空間単位 (ボクセル) から構成される。ボクセルとは、ピクセ

ルの概念を 3 次元に拡張したものであり、3 次元空間を格子状に分割して得られる単位直方体を指す。各ボクセルは、着目している属性 (材質, 温度, 密度, など) の、ボクセルの中心点におけるサンプリング値を保持する。ボクセルへの空間分割の大きさは通常は表示の解像度に対応している。

ボリュームレンダリング処理は、3 次元のボリュームデータを 2 次元に射影する画像生成処理と、その射影結果に陰影をつけるシェーディング処理に分けられる。シェーディング処理では、法線を計算できるような幾何情報が必要であるが、ボリュームデータには、表面情報がないために簡単に法線データを得ることができない。そのため、仮想的な表面の勾配をボリュームデータから再生する特別な技法が必要であり、*ReVolver*/C40 では、depth gradient shading という技法を用いて、これを実現する。

本稿では、*ReVolver*/C40 においてシェーディング処理と画像表示を行う Shading Stage について述べる。第二章においては背景として *ReVolver*/C40 の特徴と構成について述べ、第三章では *ReVolver*/C40 における Shading Stage の概要について述べる。第四章では Shading Stage において入力インターフェースを担当する Pixel Transceiver の処理、構成、実装および処理速度について述べ、第五章では Shading Stage における出力インターフェースを担当する Video Signal Generator の処理、構成、実装および処理速度について述べる。

2 *ReVolver*/C40 の特徴と構成

2.1 *ReVolver*/C40 の特徴

ReVolver/C40 は、科学技術計算の解析や流体の可視化をも行うために、高速描画、半透明ボリュームの表示、遠近法による画像生成という三つの項目を実現するように設計されている。ボリュームを半透明なものとして扱うには、図 1のようにピクセル値を計算する際に視線と交差する無色透明でない全ボクセルを順にアクセスする必要がある。また遠近法を採用する場合、任意方向の視線が発生するため、メモリアクセスが規則的ではなく、メモリバンクへのアクセス競合の可能性が高くなり、並列処理への障害となる。ここでは、それらの問題を解決し、高速描画を行うために *ReVolver*/C40 が持っている特徴を述べる。

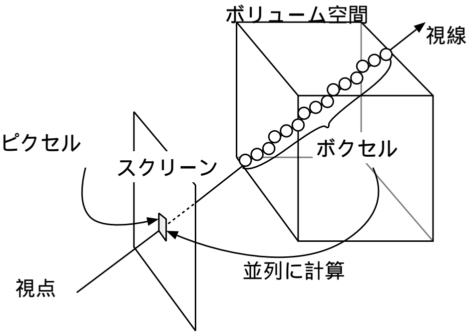


図 2: レイキャスティング法

レイキャスティング法の採用 視点からスクリーン上のあるピクセルの中心へと向かうベクトルのことを視線ベクトルと呼び、視点を一端としピクセルの中心を通る半直線を視線と呼ぶ。図 2に示すようにレイキャスティング法ではスクリーン上のピクセルごとに、視線がどのボクセルを通過してきたのかを追跡して、ピクセルの色や明るさを決定し、スクリーンに投影する。

サンプリング方法の単純化 *ReVolver*/C40 では、視線ベクトルの成分の絶対値のうち、最大値をもつ座標軸を主軸と定め、この主軸の方向に対して等間隔にボクセルをサンプリングする方法を採用している。この方法を主軸等間隔サンプリング法と呼ぶ。

バンクコンフリクトのないメモリ構成 主軸等間隔サンプリング法では、ある座標軸に垂直な平面にボリュームデータを分割してメモリバンクに格納すれば、その座標軸を主軸として持つ視線に関してはバンクコンフリクトフリーとなる。しかし、任意方向の視線の主軸が、この座標軸を主軸として持つとは限らない。そこで、x,y,z 軸のいずれにも垂直な平面が存在するように分割すれば任意方向の視線に対してバンクコンフリクトフリーとすることができる。

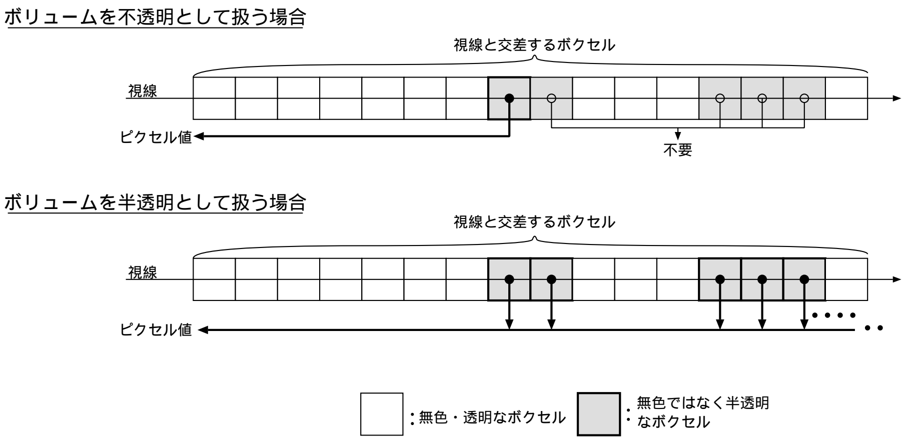


図 1: 参照するボクセルの違い

ReVolver/C40 の 3 次元メモリ構造では、ボリュームデータを三重化し、主軸等間隔サンプリングを行うことにより、視線が任意方向であっても、バンクコンフリクトが生じないことを保証する。

並列処理による高速化 *ReVolver*/C40 は、全体として RCS、PCS、SS という 3 個のステージによるパイプラインをとっている。

また、RCS では与えられた情報をスクリーンを分割して各 PE に割り当て、視線生成処理を行う。PCS ではピクセル値計算をパイプライン処理で行う。SS ではスクリーンを分割して各 PE にピクセル値を割り当て、シェーディングを行う。このように各ステージでも様々な並列処理を行うことによって、高速化を図っている。

2.2 *ReVolver*/C40 の構成

ReVolver/C40 は図 3に示すように、Ray Casting Stage(RCS)、Pixel Calculation Stage(PCS)、Shading Stage(SS) と呼ばれる三つのステージによって構成されており、これらが独立に処理を行うことによって、並列処理を実現している。この三つのステージをまとめて制御しているのが System Control Unit(SCU) である。

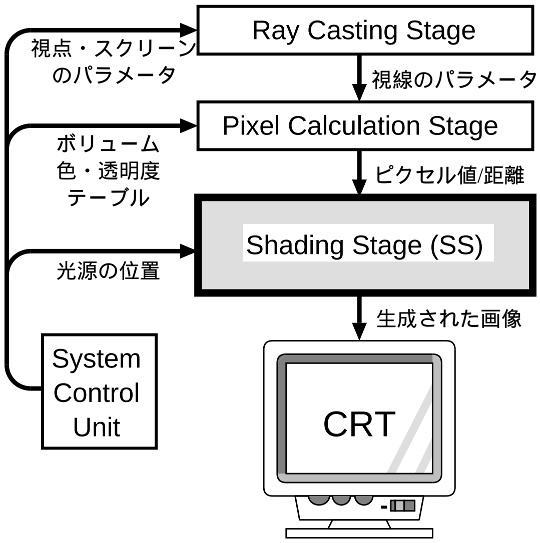


図 3: *ReVolver*/C40 の全体構成

System Control Unit システム全体の制御を行い、ホストマシンからの指示にしたがって、他の構成要素に対してコマンド/データを転送する。

Ray Casting Stage スクリーン上の全てのピクセルについて、視点からピクセルを通過する視線ベクトルを生成し、PCS へ送る。

Pixel Calculation Stage ボリュームデータを保持しているステージであり、RCS からの視線データに基づいてボクセル値を読み出す。ボクセル値は 1byte のインデックス値なので、

これを用いて属性テーブルを参照し、ピクセル値計算を行う。そしてそのピクセル値や距離を SS へ送る。

Shading Stage PCS から送られてきたデータをもとに、16 分割された PE の内の適切な PE にデータを送り、シェーディングを行う。そして CRT に出力する。

3 シェーディングステージ

3.1 シェーディング処理

シェーディングとは、立体を構成する各面に対して、光源位置や周囲の環境条件を考慮して明るさを決定する方法である。シェーディングにより、より現実的な立体の表現が可能になる。

立体を構成する各面の明るさの決定には、光源の位置や周囲の明るさ、立体の面の角度などの多くのパラメータが関与している。図 4はもっとも一般的な陰影の決定方法である。物体上のある一点 P の明るさ (輝度)L を、無限遠点にある光源からの入射光の散乱による輝度 L_d、入射光正反射による輝度 L_s、立体内部からの透過光による輝度 L_b、および周囲環境の散乱光による輝度 L_eの四つの要素の和として求める。

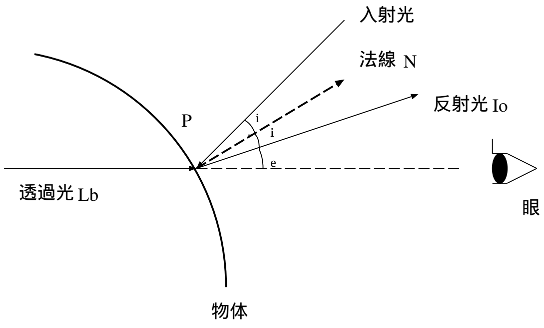


図 4: 一点 P のシェーディング

$$L = L_d + L_s + L_b + L_e \tag{1}$$

このような、物体のある点の輝度を求めることにより、シェーディングモデルが作成される。

無限遠点にある光源から強度 I_pの光が点 P に入射する場合、点 P における物体表面の法線と入射光との角度を i、対象とする立体の反射係数を R_p とすると、入射光の散乱による輝度 L_dは

$$L_d = R_p I_p \cos i \tag{2}$$

で与えられる。[7]

3.2 depth gradient shading

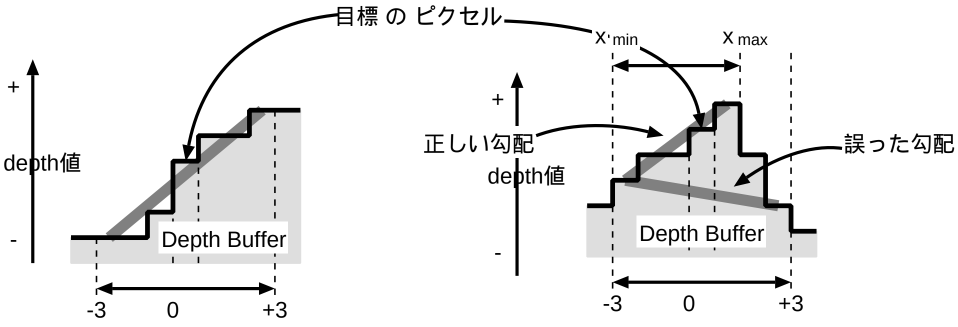
ボリュームレンダリングにおいては、ボリュームデータそのものは表面の法線ベクトルに関する情報を持たないため、周りのボクセルの状態から法線情報を推定する必要がある。この法線情報を求める方法として、*ReVolver*/C40 は depth gradient shading を採用している。ボリュームレンダリングにおいて、ボリュームの表面とは、視線上のボクセル値を視点に近い側から調べ、はじめに無色透明でないボクセルと交差した所であるとみなすことができる。

いま、スクリーンからこの表面までの距離を各ピクセルごとに求め、2 次元配列に格納したものを深さバッファ(depth buffer) と呼び、深さバッファに格納される値を depth 値と呼ぶ。この depth 値から表面の法線を近似計算し、ピクセル値と法線ベクトルをもとに各ピクセルの輝度を補正する手法が、depth gradient shading である。

ある点 P における法線ベクトルを求める場合、X 方向については図 5の (a) のように depth 値が単調増加する場合, 斜線区間の両端ピクセルについて depth 値を取り出し、depth 値の差と座標の差を使って勾配を近似する。また、図 5の (b) のように depth 値の増加が変化する場合は、(a) の場合と同様に差を取ると勾配が平坦になり突起が再現されないので、近似に用いる区間を選択する。Y 方向についても同様に勾配を近似し、2 つの勾配から法線ベクトルを求める。

3.3 depth gradient shading の並列化

ReVolver/C40 の行う depth gradient shading は、図 6 に示した求めるピクセル自身を含む上下左右合わせて 13 ピクセルの領域に対するデータから法線ベクトルを求めて輝度を計算する。この際、各ピクセルの計算は相互に依存が無いため、ピクセル単位のデータ並列処理が可能である。一方、法線計算におけるデータのアクセスパターン



(a)depthが 単調 増加/ 減少する場合　　(b) depthの 増減が変化する場合

図 5: 法線ベクトル

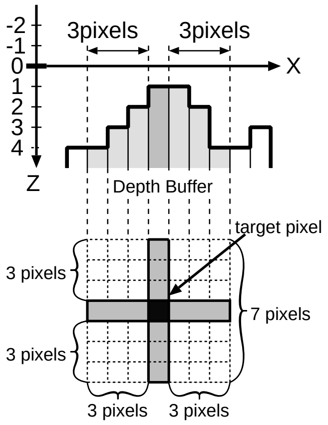


図 6: depth gradient shading の法線計算に必要なピクセル

には、x 方向、y方向の両方向に関して、参照の局所性が存在する。したがって、効率的な並列処理を行うためには、1 次元、あるいは 2 次元のブロック分割が適している。そこで、*ReVolver*/C40 では画面を図 7 のように 2 次元ブロック分割 (4×4 の 16 分割) し、それぞれを一つの PE に割り当てることで、並列処理を行う。

3.4 シェーディングステージの構成

Shading Stage は大きくわけて Pixel Transceiver、Shading Unit、Video Signal Generator の三つの部分からなっている。

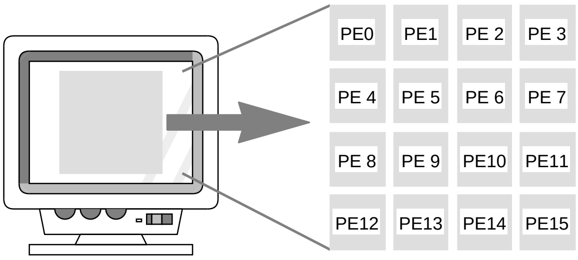


図 7: フレーム ・バッファの分割

Pixel Transceiver 各ピクセルのデータを前のステージであるピクセル値計算ステージから受けとり、そのピクセルの座標をもとに 16 分割された PE のうち、どの PE の担当であるかを計算し、その PE にデータを渡す処理を行う。

Shading Unit 16 台の Processing Element(PE) の集合体であり、分散フレームバッファを構成する。Pixel Transceiver から受け取ったデータをもとに、depth gradient shading によるシェーディング処理を施し、その結果を分散フレームバッファに書き込む。

Video Signal Generator 各 PE の分散フレームバッファから、RGB データを適切な順序、かつ適切なタイミングで収集し、受け取った RGB データ (ディジタル信号) を D/A Converter を通してアナログ信号に変換し、ディスプレイ画面に出力する。

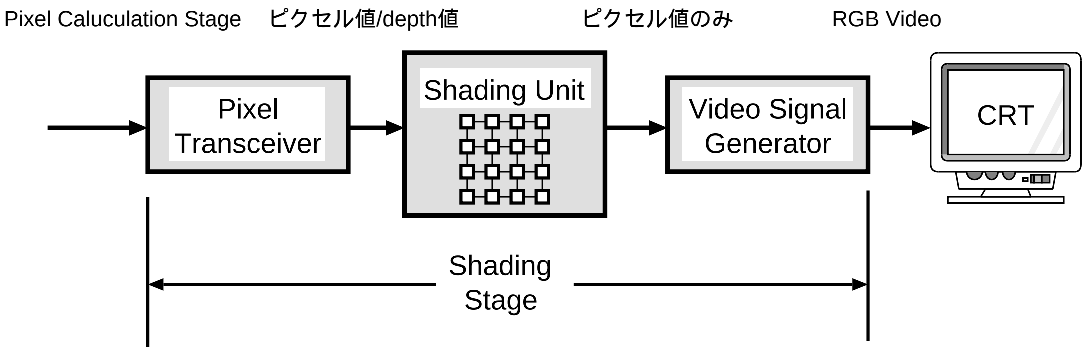


図 8: Shading Stage の全体構成

3.5 基板設計

シェーディングステージは、図 9のようにメイン基板 2 枚からなる。1 枚のメイン基板には、PE を 1 個持つサブ基板 8 枚と、そのサブ基板に一つずつ割り当てられているシリアルクロック生成用の Programmable Array Logic(PAL) が 8 個、PT を実現するための Field Programmable Gate Array(FPGA)1 個、CRT Controller を実現するための FPGA1 個と D/A Converter1 個ならびに、バス制御用のバッファなどが搭載されている。

ピクセル値計算ステージからは、それぞれの基板にバスが 2 本入力されている。また CRT には、D/A Converter から送られる RGB 値の入力バスと同期信号の入力線がある。

また、2 枚の基板は一方が Master、もう一方が Slave に設定される。PT の PCS へのデータ送信要求、VSG の PE へのデータ送信要求と CRT へのデータ送信は、Master の基板が代表して行う。

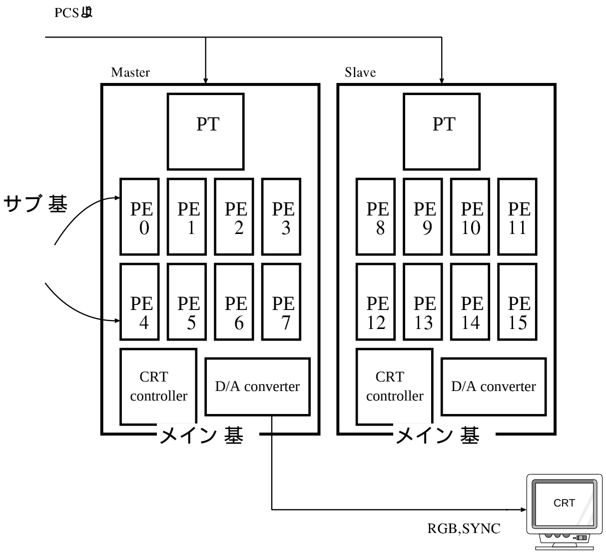


図 9: 基板設計

3.6 Shading Unitの構成

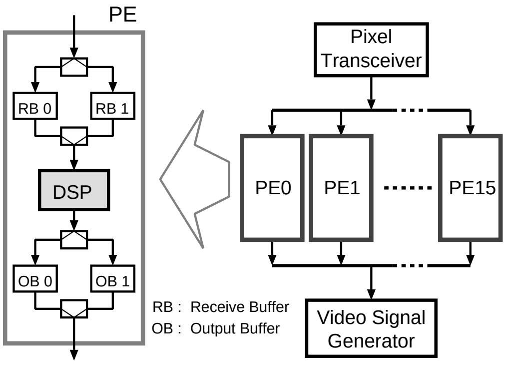


図 10: Shading Unit(SU)の構成

Shading Unitは、16台のProcessing Element(PE)を格子状に配置し、各Digital Signal Processor(DSP)が持つCom Portによりメッシュ結合した構成となる。ただしPE0とPE3、PE4とPE7、PE8とPE11、PE12とPE15については、リング結合とする。PEの構成は図10に示すように、depth値の計算とその結果を用いてシェーディング処理を行うDSPを中心に、PTがPCSから受け取ったピクセル値やRGB値などのピクセルデータを書き込むためのReceive Buffer(RB)と、シェーディング処理を終えたRGBデータを格納するOutput Buffer(OB)で構成されている。OBはCRTに出力するためのフレームバッファの役割もかねている。これを一台のPEとし、16台のPEで並列処理を行う。

図10のようにRBに対するPixel Transceiverからの書き込みとDSPからの読み出し、OBに対するDSPからの書き込みとVideo Signal Generatorへの転送という処理を並列に行うため、RBとOBはそれぞれ2バンク構成にし、フレーム単位のシェーディング処理が終わるたびにPixel Transceiverがバンクを切り換える。RBは図11に示すように、Pixel TransceiverとDSPの2ヶ所からアクセスされる。RBのアクセスの特徴としては、次のようなものがある。

- Pixel TransceiverとDSPのアクセスは、全く異なる領域に対して行われる。
- Pixel Transceiver側もDSP側も散発アクセスである。

図11に示すように、PTがRBにアクセスする時間は短く、またPEは16個あるのでPTが1個のPEのRBを占有する時間は僅かである。よってRBはSRAMを2個接続し、物理的には1ポートで実現され、時分割で仮想的に2ポートにしている。そしてPTからのアクセスがあった場合にそちらを優先させるようにしている。

また、OBも図11に示すように、DSPとVideo Signal Generatorの2ヶ所からアクセスされる。OBのアクセスの特徴としては、次のようなものがある。

- DSPとVideo Signal Generatorのアクセスは、全く異なる領域に対して行われる。
- DSP側は散発アクセス、Video Signal Generator側は単発の連続アクセス。

図11に示すように、VSGがOBにアクセスする時間は長く、その間OBを占有してしまうとDSPが長い間OBに書き込むことができなくなる。よってOBではRAMポートとSAMポートという2つのアクセスポートを持ったVRAMを使用している。ここでは、DSPからの書き込みには高速性は要求されていないので、DRAMと同じ方法でアクセスできるRAMポートを使用し、Video Signal Generatorへのデータ転送は、高速なシーケンシャルアクセスが可能なSAMポートを使用する。

4 Pixel Transceiver

4.1 PTの要求仕様

ReVolver/C40は、視線生成/ピクセル値計算/シェーディング処理/という3つの処理を、図12に示すようにRCS/PCS/SSという3つのステージでパイプライン処理を行っている。PCSはピクセル値計算をパイプラインで処理しており、$1520ns$で1ピクセルの処理を終え、データをSSに送る。[4] Pixel Transceiver(PT)には図13に示すようにPCSから送られてくるデータを、PCSのパイプラインを乱すことなくSUの16個のPEに分配することが要求される。そのためにPTは、$1520ns$以内に1つのピクセル値をPCSから受け取り、適当なPEに渡さなければならない。

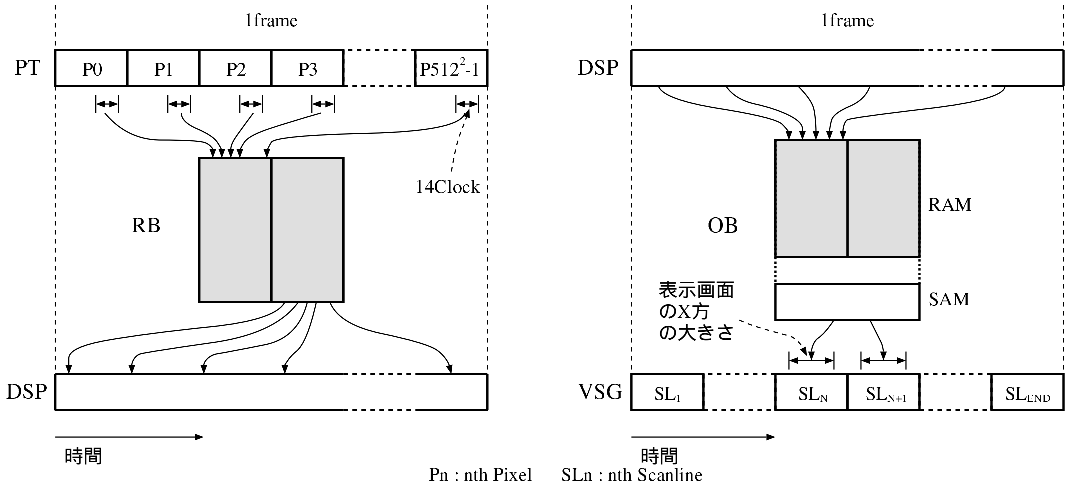


図 11: RB と OB のアクセス方法の違い

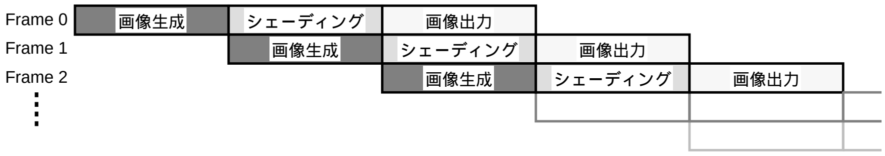


図 12: *ReVolver*/C40 のパイプライン処理

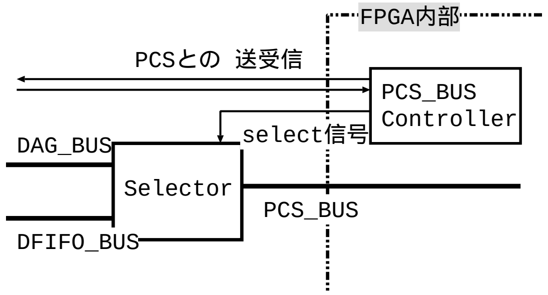


図 15: PCS と PT を流れる 2 本のバスの制御

4.2 PT の処理と構成

各ピクセルのデータを前のステージであるピクセル値計算ステージから受けとり、そのピクセルの座標をもとに 16 分割された PE のうち、どの PE の担当であるかを計算し、その PE のもつメモリ (Receive Buffer) に受け取ったデータを書き込む。そのためのアドレス計算もする。

PT の全体構成を図 14に示す。

Load Controller PT 全体にわたるグローバルな制御としては、大きく以下の 3 つの制御があり、これらの制御を担当しているのが Load Controller である。

- PCS_BUS Controller を介した、PCS との送受信を制御。
- Shading Unit 内の Receive Buffer へのアクセス制御。SU とのハンドシェーク、データバス (DATA_BUS) とアドレスバス (ADDRESS_BUS) の制御。
- アドレス計算の制御。

PCS_BUS Controller PCS からのデータ受信の制御を行っているのが、PCS_BUS Controller である。図15のように、PCS と接続されているバスは 2 本あるが、PCS_BUS Controller で FPGA に入るバスを 1 本選択する。

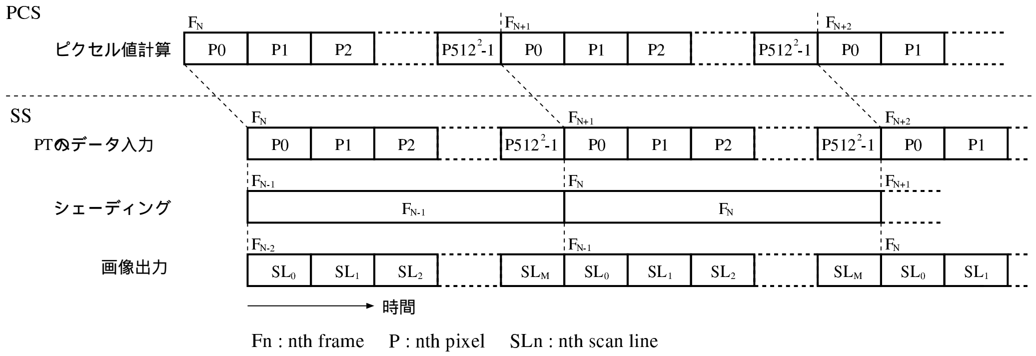


図 13: SS のパイプライン処理

Float Controller RGB 値は PCS から、それぞれ 32 ビットの浮動小数点数で送られてくるが、後述の VSG で画像表示する際に D/A コンバーターに送られるのは、RGB それぞれ 8 ビットの整数なので、Float Controller でこの変換を行う。

SU_BUS Controller PCS から受け取ったデータは 16 個の PE に配分される。ピクセルの座標値に従って、PE を選択し、各 Receive Buffer にデータを書き込むためのアドレス計算を SU_BUS Controller で行う。

4.3　PT の実装

ここでは PT の実装、特にアドレス計算の実装と各種のタイミングについて述べる。

4.3.1　アドレス計算

PCS から受け取ったデータは、PE の Receive Buffer に書き込まれる。RB 内部のアドレス計算は、前に述べたように SU_BUS Controller で行われるが、そのアドレス計算は、以下のように行われる。ここで、addr1、addr2 を 3 ,4の式で求める。

$$addr1 = y \times 128 + x \tag{3}$$

$$addr2 = (y + 4) \times 128 + (y + 4) \times 8 + x + 4 \tag{4}$$

addr1 は yを 7 ビットシフトするだけで求められるが、addr2 は様々な実装方法がある。この加算には、4 個の加算が存在するので、図 16のように普通 4 個の加算器で構成されると考えられる。

図 17(1) の実装方法だと、4 個の加算器がすべて 8 ビットの加算器で構成することができる。これは、高速化や FPGA 内で占める領域の小ささの点でメリットがあるといえる。また、図 17(2) ような実装方法もある。この方法では加算器が少ない点で優れていると言えるが、配線量の多さが問題である。現在は、見た目にも容易に実装できる 1 の方法をとっている。

この addr1,addr2 の値を用いて式 5,6,7,8 のようにそれぞれのアドレスを求める。

$$RGB\text{値のアドレス} = addr1 \tag{5}$$

$$\text{交点座標のアドレス} = (addr2 \div 2) + 4000H\text{[1]} \tag{6}$$

$$\text{初期距離のアドレス} = (addr2 \times 2) + 6800H \tag{7}$$

$$\text{基本距離のアドレス} = (addr2 \times 2) + 6801H \tag{8}$$

4.3.2　各種タイミング

データ受信準備 図 18の S0 から S5 の間では、PCS からのデータを受け取る準備を行う。Load Controller からのピクセルデータを受け取る許可の信号が来るのを待つ。そして RDY 信号により、DAG_BUS(AG バス) からのデータ送信の準備が出来たことを PCS が SS に伝える。また、Master と Slave の PT が、両方準備ができるまで待つ。

DAG_BUS からのデータ受信 S6 から S11 の間では、SS から PCS に AG バスからのデータ

---

[1] H は 16 進数を表わす

表 1: DAG_BUS を流れるデータ

順番	データ名	解説
1	IN1,BV1	ボクセルの入射点の y 座標と基本ベクトル。
2	IN2,BV2	ボクセルの入射点の z 座標と基本ベクトル。
3	FRAME,MASK	フレームビット, サンプリング開始位置
	AXIS,DIR	主軸名, 主軸方向
	PXL.x,PXL.y	スクリーン座標をあらわす。
4	BLEN	基本距離 (視線ベクトルの基本距離) を表わす。
5	ILEN	初期距離 (視点から、視線ベクトルがボリューム空間に入射した点までの距離) を表わす。

表 2: DFIFO_BUS を流れるデータ

順番	データ名	解説
1	DIR、INTS	主軸方向, 交点座標。(視線から一番近い無色透明でないボクセルを表わす。)
2	R	ピクセル値の赤成分。(float)
3	G	ピクセル値の緑成分。(float)
4	B	ピクセル値の青成分。(float)

の送信要求をだす。そして、流れてきたデータを受け取る。AG バスでは図 19と表 1に示すようなデータが送られてくるので、それを各ラッチに取り込む。このデータは、一度送信要求を出すとクロックに同期して 5 つ連続して流れてくる。

アドレス計算 S10 から S12 の間で SU 番号を選択し、addr1,addr2 の値を計算する。その後、addr1,addr2 の値を用いてそれぞれの RB のアドレスを計算しそれぞれのアドレスラッチに格納する。

DFIFO_BUS からのデータ受信 PCS は Empty Flag 信号によって、
DFIFO_BUS(DSP バス) からの準備ができたことを SS に伝える。DSP バスからは図 19と表 2に示すようなデータが送られてくる。S14 から S26 の間では、SS は Empty Flag が 0 のときに PCS へのデータ送信要求を行い、データが送られてくると RGB 値に関しては型変換を行って (32bit-float → 8bit-integer)、それぞれのラッチに格納する。DSP バスからのデータは、一度のデータ送信要求に対し一つずつしかデータが流れてこないので、データを受け取るたびに、Empty Flag が 0 になったことを確認してからデータの送信要求を出す。

RB へのデータ送信 S16 から S29 では、PT で計算されたアドレスをもとに、RB へのデータの書き込みを行なっている。RB のアドレスバスとデータバスは、DSP と PT で共有されている。このため PT と DSP の 2ヶ所からなされるアクセスは互いに排他的でなければならない。S16 で SU_SEL 信号が 1 になると、RB の制御を行っている RB Controller は、PT からのアクセスを優先しデータ受信フェーズに入る。受信可能になるまでに 2 フェーズ、データ受信に 3 フェーズ消費する。その後 SU_SEL 信号が 1 の間は、RB は連続してデータを受け取ることができる。RB がデータを受け取ると RB Controller は ACK を返し、データを受け取ったことを PT に伝える。[2]

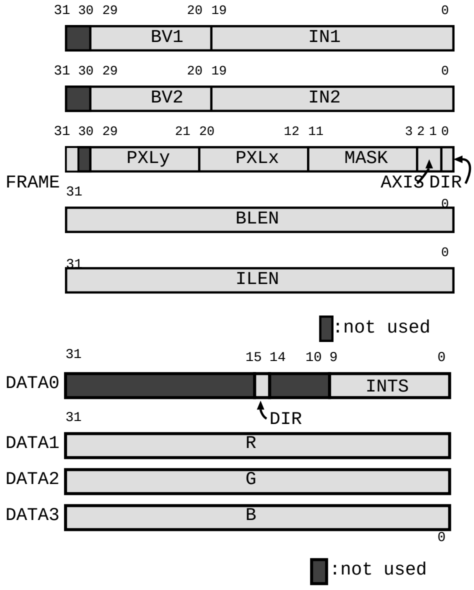


図 19: Pixel Calculation Stage から受け取るデータのフォーマット

4.4 PT の処理時間に関する考察

この節では、PT の実装上の処理時間とそれに関する考察を述べる。

4.4.1 PT の一つのピクセルの処理時間

PT で使用するクロックの周波数は、25MHz である。SCU で生成されて SS に送られてくるクロック周波数は、12.5MHz なので、これを利用して SS で 25MHz のクロックを生成する。現在のところ PT では、図 18に示すように、1 ピクセルにつき 31 クロック消費する。

4.4.2 データ分配方式に関する考察

図 20のようにフレームを分割したとき問題となるのが、一つの PE が担当する領域の内、上下左右 3 ピクセルの扱いである。図 20 に示すように PE が自分の担当する領域のシェーディングを行うとき、上下左右 3 ピクセルについては自分の領域にない depth 情報を用いて法線を計算する必要がある。この問題を解決するための方法として、

1. 各 PE 上に上下左右 3 ピクセル分の周辺領域の depth 情報を一部重複して割り当てる。
2. DSP 間通信によって、ピクセル値を持つ隣接 PE から depth 情報を取得する。

がある。

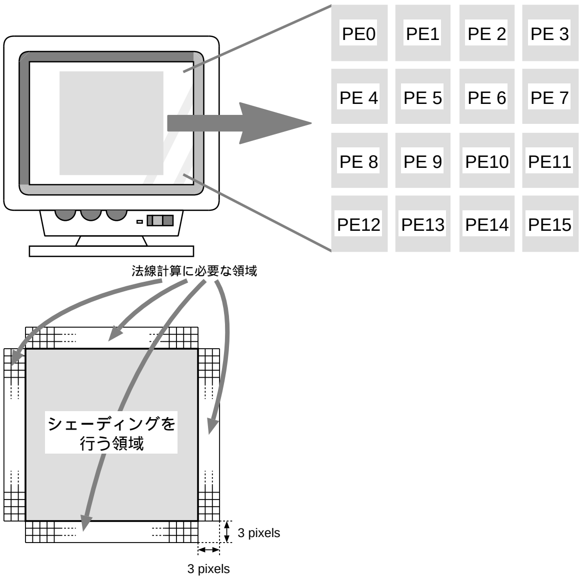


図 20: フレーム・バッファの分割と法線計算に必要な depth 情報の範囲

PCS は、4 フレームを 1 個の DSP で処理しており、その場合、PCS が 1 ピクセルを処理するのに最高 $1520ns$(38 クロック) でできると予想されている。[4]

1 の方法では、共有されているピクセルのアドレス計算まで行うと、そのアドレスへデータを書き込む回数が、最大 3 倍になるので、38 クロックで処理を終わらせるのは不可能である。パイプラインをとることも考えられるが、1 フレームごとにデータがまとめて送られてくるので、フレーム単位のパイプラインを構成するのは難しく、高速化を図るのは困難である。PT で処理が遅れると前のステージの動きを止めることになり問題である。

2 の方法では、シェーディングがどの程度の期間で終わるかが問題になってくる。PT が RB に書き込みを行っている間は、SU では RB を読むことができないので、図 18より PT が 1 ピクセル当りに RB にアクセスする平均時間は $14 \times 40 \div 16 = 35ns$、アセンブリ言語で depth gradient shading を DSP で行うプログラムを書くと、X、Y 方向それぞれの勾配の計算に 41 命令、ピクセルのロード、ストアを含めた輝度の計算に 69 命令、合計

151 命令必要である。[3] 一方、DSP の命令サイクルは $40ns$ であるので、DSP 間通信を行わない場合、1 ピクセルのシェーディングに必要な処理時間は $6040+35=6075ns$ である。また、DSP 間の通信時間も影響してくる。DSP の Com Port を使用した場合、最大で $20Mbyte/s$、すなわち 1word につき $200ns$ で通信が可能である。DSP 間通信を行うとき、送信側と受信側で 1 命令ずつすなわち 1 サイクルずつ、バスを用いたデータの書き込みに 2 サイクル、書き込みアドレスの計算に 15 サイクル消費する。[13] そして 1 ピクセルのデータは 4word なので、1 ピクセルのデータを DSP 間通信で受け取る時間は $(19\times40+200)\times4=3840ns$ である。よって DSP 間通信を行った場合の 1 個のピクセルの平均シェーディング時間は、$(6075\times128\times128+3840\times128\times12)\div(128\times128)=6435ns$ である。ここで、SU が 1 ピクセルのシェーディングにかけられる最大時間を求める。SU では 1 フレームを 16 個の PE で並列処理を行うので、$1520\times16=24320ns$ で 1 ピクセルのシェーディングを行えば良い。2 の方法では、十分これを満たしている。よって PT では 2 の方法を採用した。

この方法で PT は現在、すべての処理を 31 クロックで終えることができるので、PT も全体のパイプラインを止めることなく処理を終えることができる。

5 Video Signal Generator

5.1 VSG の要求仕様

4.1節でも述べたように、*ReVolver*/C40 はパイプラインをとっている。Video Signal Generator(VSG) でもそのパイプラインを乱さないように OB から RGB データを受け取り、その RGB データをデジタル信号からアナログ信号に変換し、画像を出力することが要求される。

5.2 VSG での処理と構成

VSG では、SU でシェーディング処理を終えて、OB からシリアルクロックに基づいて送られてくるアナログの RGB データをデジタル信号に変換し、CRT に出力する。CRT に出力する時の走査の方法は、一画面を一度にすべて走査するノンインタレース走査である。この方法は、パソコンなどの表示に使用され、通常のテレビに採用されているインタレース走査にくらべると、ちらつきが少ない。[5]

VSG の全体構成を図 21に示す。

CRT Controller VSG で中心的な役割を果たしているのが CRT Controller である。主な処理として、スイッチによる各種の値の入力制御と制御信号生成、画像生成のための各種信号の生成、画像データ出力の制御がある。

CRT Controller については、5.3、5.4節で詳しく述べる。

D/A Converter VSG で使用する D/A Converter は、3 個の独立した 8 ビットの RGB データ入力ポートと、ビデオ入力制御用のコンポジット同期信号、ブランクなどの入力ポートを備えている。D/A Converter では、これらの信号を受け取り、RGB データをアナログ信号に変換し、CRT に出力する。

Serial Clock Generator Output Buffer の SAM ポートに送るシリアルクロックは各 Processing Element に割り当てられている Programmable Array Logic(PAL) で生成されている。シリアルクロックの送信のタイミングは、CRT Controller によって制御している。またこのシリアルクロックを生成する PAL は 50MHz のクロックで動作している。

Phase-Locked Loop(PLL) Phase-Locked Loop(PLL) は入力信号に位相同期した発信出力が得られる位相同期回路である。VSG ではこれを用いて、SCU から送られてきた 12.5MHz のクロックから VSG のほとんどの部分で使用する 25MHz のクロックと、Serial Clock Generator で使用する 50MHz のクロックを生成する。

5.3 CRT Controller の構成

CRT Controller では SU からの RGB データを受け取り、D/A Converter に渡すという処理を行

う。そのとき Video Data Bus を介してデータが流れてくるが、この Bus は基板 1 枚につき 1 本あるために、どちらの基板から流れてくるデータなのかを判定しなければならない。

CRT Controller の全体構成を図 22に示す。

ここで CRT Controller を、それぞれの働きから図 22のように 3 つの部分にわける。

Timing Signal Generator ここでは、Display Signal Generator と SU Controller のタイミングの制御を行っている。Timing Signal Generator には 4 個のカウンターと 6 個の比較器があり、これらを用いて各種制御信号を生成している。

Display Signal Generator ここでは、Timing Signal Generator からの制御信号を用いて、コンポジット同期信号とブランク信号を生成する。

SU Controller ここでは、OB の中の RAM ポートから SAM ポートへのデータ転送の制御と、D/A Converter へのデータ転送制御と、その際の Video Data Bus の選択の制御を行う。

5.4 CRT Controller の動作

ここでは、CRT Controller の動作について述べる。

5.4.1 ディスプレイへの描画制御

CRT の表示画面は、図 23のようになっている。そこで、Timing Signal Generator には、スイッチによって次の値が設定される。

- 表示画面の初期座標
- 表示画面の大きさ
- 全走査線数と全走査幅数

スイッチで設定された値をそれぞれのラッチに格納する時の制御は、Switch Controller で行われる。

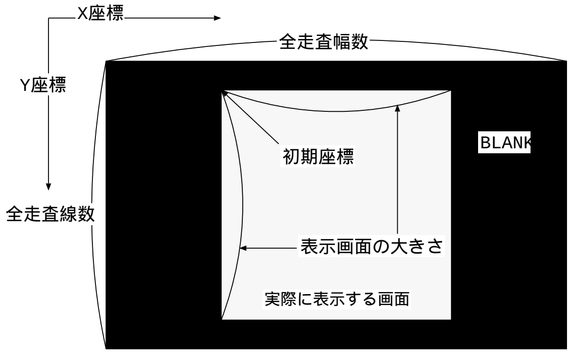


図 23: 表示画面

コンポジット同期信号の生成 画像信号を伝送して、元の画像を正しく CRT に再現するためには、データを送る側と受ける側の両方で正しく同期をとる必要がある。[5] そこで、2 次元画面を構成するための同期信号として、水平同期信号と、垂直同期信号がある。水平同期信号は各ラインの先頭にあり、垂直同期信号は、各フィールドの先頭にある。これらは、画面上で画像として現れないブランキング期間に付加されている。これらの同期信号は別々に送られることもあるが、通常はこの二つを複合してコンポジット同期信号として伝送し、受け取る側で水平同期信号と、垂直同期信号に分離する。ただし、水平同期信号に垂直同期信号を単純に加算したのでは、受ける側で分離することができないことも予想されるため、垂直同期信号部分では図 24ような加工がなされている。

垂直同期信号期間は、3 ライン分の垂直同期信号と、その前後に 3 ライン分の等価パルス期間の組合わせで合計 9 ライン分が各フィールドにある。これによってコンポジット同期信号からの水平垂直各同期信号の分離を容易にする目的がある。

ブランク信号の生成 画像信号には、ブランキング期間と呼ばれる実際の画像信号の存在しない場所がある。[5] (図 23参照) この部分のディスプレイの場所を知らせる信号であるブランク信号を生成する。

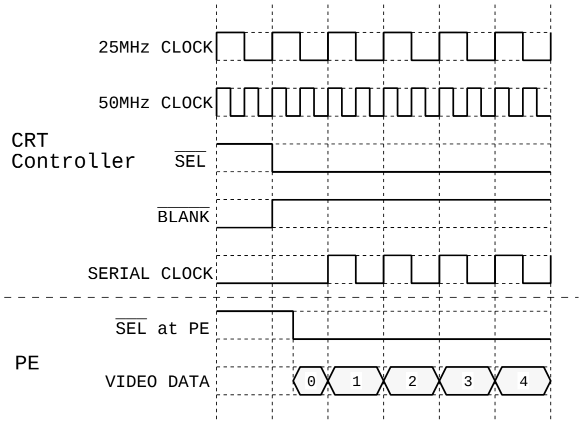


図 25: シリアルクロック生成と D/A Converter へのデータ転送のタイミング

5.4.2 分散フレームバッファからのデータ読み出し制御

表示すべきピクセルの座標を計算し、そのピクセルをもつ SU の OB に対し転送するピクセルの row アドレスをわたす。データ転送要求信号と row 信号がともに Active になると PE の VRAM 内で RAM ポート内にある 128 個のデータが SAM ポートに転送される。このデータ転送要求信号の制御は、DT_REQUEST Controller で行われる。転送完了後 SAM ポートはシリアル出力モードに切り替わる。そして col 信号が Active になると一つの PE が選択され、シリアルクロックの立ち上がりに同期して、SAM のスタートアドレスからのデータをシリアルに出力する。Serial Clock Generator でシリアルクロックを生成する際、50MHz のクロックを使用する。Serial Clock Generator は、図 25に示すように、$\overline{SEL}$信号と 25MHz のクロックがはじめて共に Low になったときに、シリアルクロックを生成しはじめる。このとき、座標を数えるカウンターで数えている座標のデータを送らなければならないので、2 クロック前に PE が選択されていなければならない。この制御は非同期で行われるため、データの流れを正しく理解することが重要である。

5.4.3 D/A Converter へのデータ転送のタイミング

OB から出力される RGB データは、それぞれの PE が持つ Video Data Bus を通る。Video Data Bus には、PE の外側にラッチが付加されており、選択信号で PE が選ばれると、SU Controller でラッチを開き、表示画面用のカウンターで数えられている座標の RGB データを、出力しなければならない。VSG には Master と Slave が存在するが、両方の CRT Controller を同時に動かすことはできない。よって一方が動いていればもう一方は動かないような制御が必要である。そのため、Slave からの RGB データは基板をまたがって流れてくることになり、時間がかかる。そして、D/A Converter へ流れるバスは 1 本しかないので、どちらの Video Data Bus から流れてきたかを判断し、制御しなければならない。この制御は SEL_VDB 信号を、マルチプレクサの制御線に入力することによって行う。(図 21参照) タイミングの問題を解決するために、D/A Converter へのデータ転送のタイミングは、図 25、26のようになっている。

5.5 VSG の処理速度

1 秒間に 60 枚 (フレーム周波数 $60frame/s$) の画像を表示する場合、$1frame$ にかけられる時間は約 $16.7ms$ である。ここで、全走査幅数を X、全走査線数を Y とすると、ドットクロックが 25MHz なので、X と Y の関係は式 9で与えられる。

$$X \times Y \times 60 = 25 \times 10^6 \qquad (9)$$

よって $X \times Y \div 4.17 \times 10^5$である。また、水平同期信号をとるのに約 $10.9\mu s$、垂直同期信号をとるのに $20Line$ 必要なので、512^2で画面に表示する場合、$X = 780, Y = 535$ ほどの走査面が必要である。ここで $780 \times 535 \div 4.17 \times 10^5$であるので、$512^2$より表示画面を大きくするには、フレーム周波数を減らす必要がある。

6 おわりに

本稿において、*ReVolver*/C40 におけるシェーディングステージの構成、設計、及び実装について述べた。

シェーディングステージでは、シェーディング処理と画像出力という二つの処理を Pixel Transceiver(PT),Shading Unit(SU),Video Signal Generator(VSG) の三つの部分でパイプライン処理することによって行っている。また、シェーディングは、フレームを16分割し、それぞれの部分を1個の Processing Element(PE) が担当することにより並列処理を行っている。くわえて、シェーディングの手法には高速なシェーディング処理の可能な depth gradient shading 方式を採用している。

基板上の設計としては、シェーディングステージはメイン基板2枚で構成されており、1枚のメイン基板には、PE1 個が実装されたサブ基板が8枚と、各サブ基板に1個割り当てられているシリアルクロックを生成するための Programmable Array Logic(PAL) が8個、Pixel Transceiver 用の Field Programmable Gate Array(FPGA) が1個、CRT Controller 用の FPGA が1個、そして D/A Converter が1個搭載されている。

PT と VSG はどちらもそれぞれの基板に1個ずつあるが、外部のスイッチで Master か Slave かを切り換えることができる。

各 PE は DSP と、Pixel Calculation Stage から来たピクセルデータを書き込むための Receive Buffer(RB) と、シェーディングを終えた RGB データを格納する Output Buffer(OB) から成り立っている。RB と OB は2フレーム分のデータを持たせることで2バンク構成にしている。また RB は高速 SRAM を使用し、仮想的2ポートをとることによりデータの書き込み/読み出し処理を同時に行うことができる。そして OB は DRAM と同じ方法でアクセス可能な RAM ポートと、高速なシーケンシャルアクセスが可能な SAM ポートという2つのポートを持つ VRAM を使用することにより、データの書き込み/読み出し処理を同時に行うことができる。

今後、SU 内のシェーディングプログラムを開発する。基板ができしだいにデバッグを行い、性能評価を行う。また、画像をハイビジョンで出せるようにするために機能を拡張することも考えている。

謝辞

本研究の機会を与えてくださった、富田 眞治教授に深く感謝の意を表します。

数々の有用な御指導、御意見を頂いた中島 浩助教授、森 眞一郎助教授、五島 正裕助手に心から感謝致します。

本研究の共同研究者である、藤原 雅宏氏、木中一博氏をはじめ、計算機工学講座計算機構成分野の諸氏に感謝致します。

参考文献

[1] 木綿 芳行 ボリュームレンダリング専用並列計算機 *ReVolver*/C40 のシェーディングステージの設計, 京都大学特別研究報告書 (1996)

[2] 荻野 友隆 :ボリュームレンダリング専用並列計算機 *ReVolver*/C40 シェーディングステージ仕様書, 京都大学 (1995)

[3] 荻野 友隆 :ボリュームレンダリング専用並列計算機 *ReVolver*/C40 のシェーディングステージ, 京都大学特別研究報告書 (1995)

[4] 中山 明則 :ボリュームレンダリング専用並列計算機 *ReVolver*/C40 のピクセル値計算ステージの設計, 修士論文, 京都大学 (1996)

[5] トランジスタ技術 SPECIAL No.31 特集 基礎からのビデオ信号処理技術

[6] 鷺島 敬之, 西澤 貞次, 浅原 重夫 共著 :並列図形処理

[7] 中嶋 正之, 川合 慧 共著 :グラフィクスとマンマシンシステム

[8] XILINX : WORKVIEW SERIES 1 VOLUME 2

[9] XILINX : LIBRARIES GUIDE

[10] XILINX : VIEWLOGIC INTERFACE USER GUIDE

[11] XILINX :XILINX-ABEL SOFTWARE DESIGN

[12] XILINX : XILINX ABEL USER GUIDE

[13] Texas Instruments :TMS320C4x User's Guide

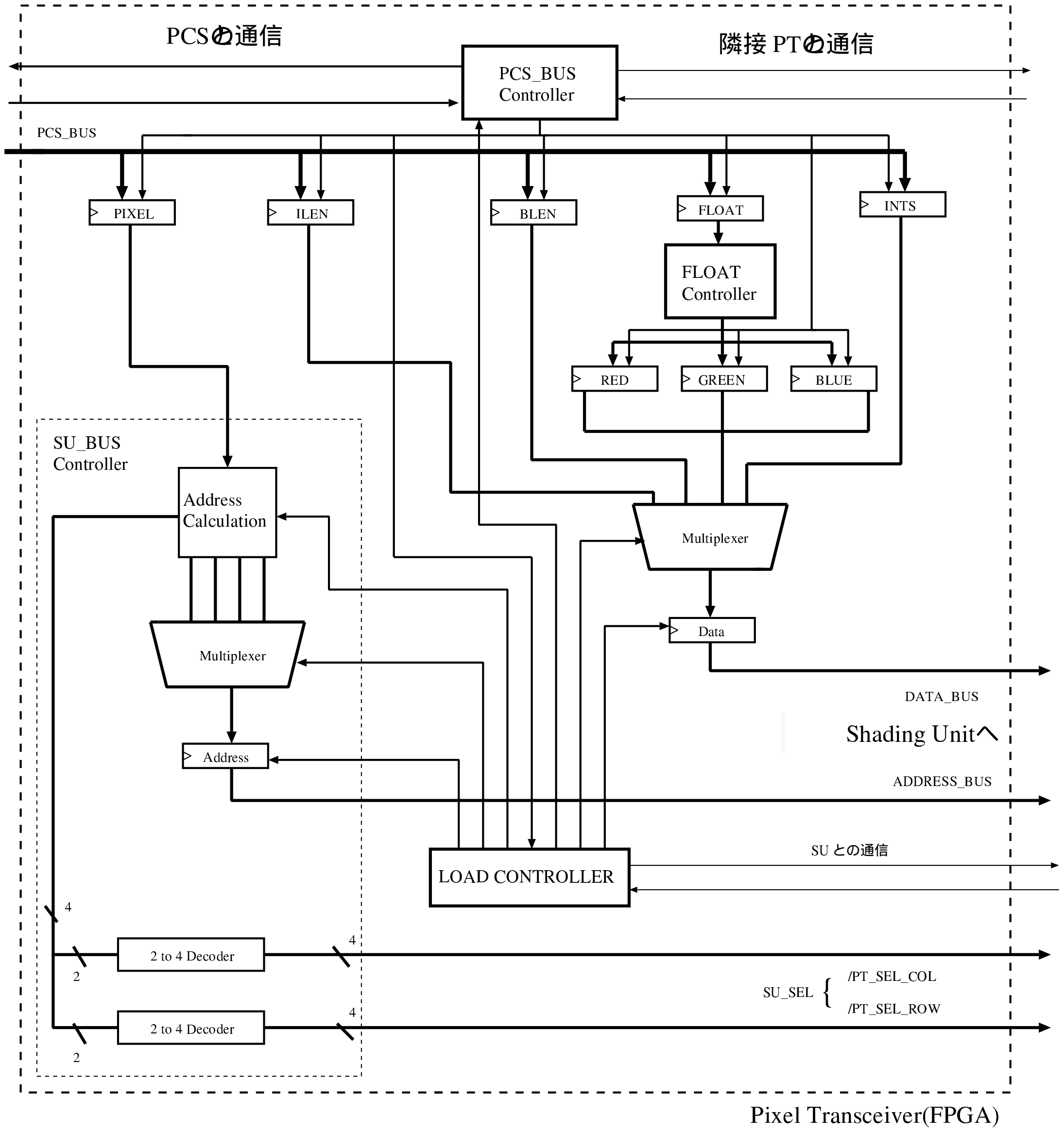

PCSとの通信
隣接PTとの通信
PCS_BUS
Controller
PCS_BUS
PIXEL
ILEN
BLEN
FLOAT
INTS
FLOAT
Controller
RED
GREEN
BLUE
SU_BUS
Controller
Address
Calculation
Multiplexer
Multiplexer
Data
DATA_BUS
Shading Unitへ
Address
ADDRESS_BUS
SUとの通信
LOAD CONTROLLER
4
2 to 4 Decoder
4
2
/PT_SEL_COL
SU_SEL
/PT_SEL_ROW
4
2 to 4 Decoder
2
Pixel Transceiver(FPGA)

図 14: PT の全体構成

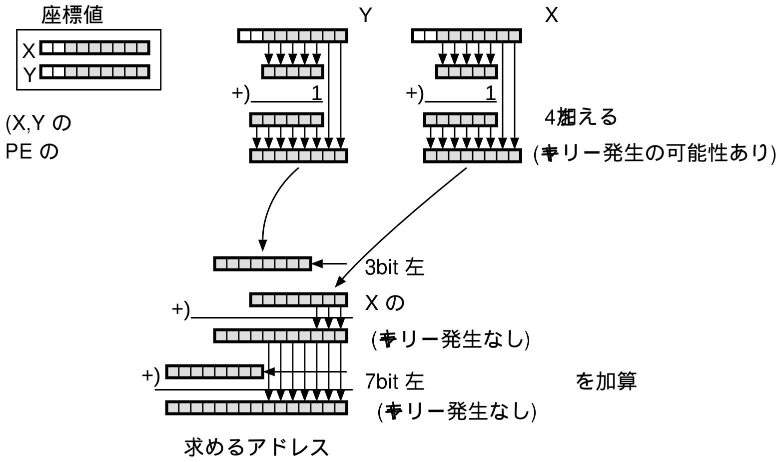
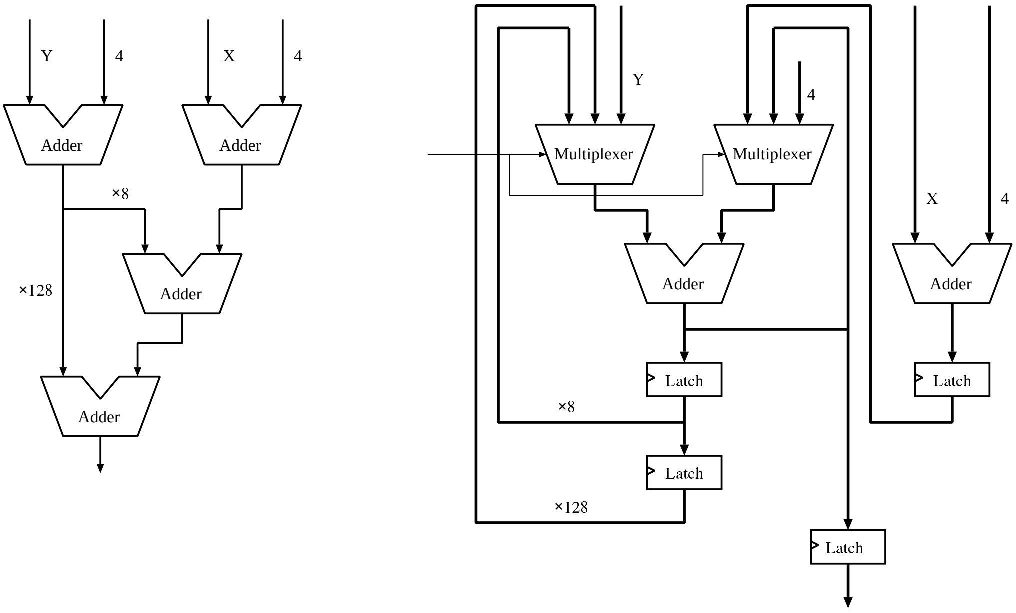


図 16: PE 内アドレスの計算方法



(1)

(2)

図 17: PE 内アドレスの計算回路

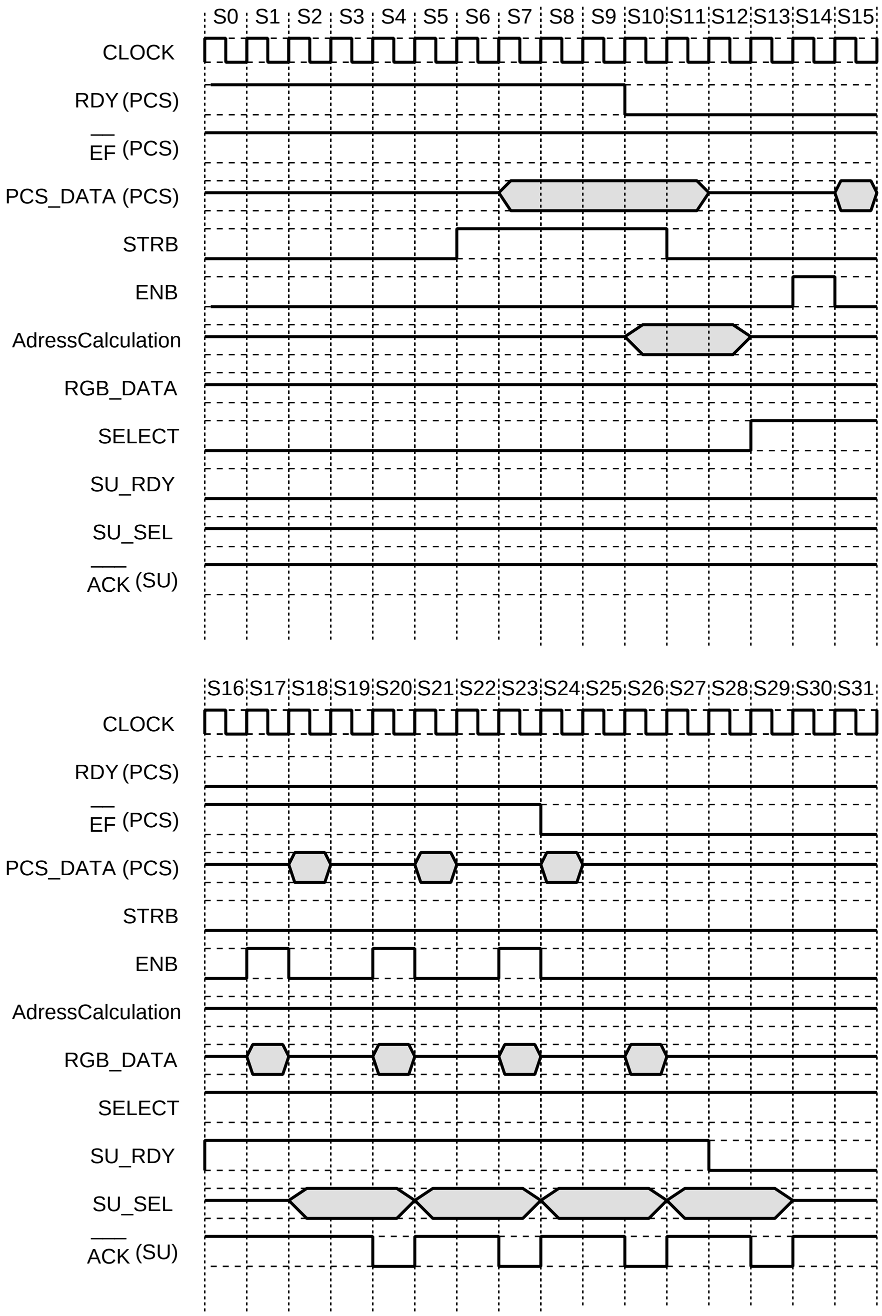


図 18: PT における各種タイミング

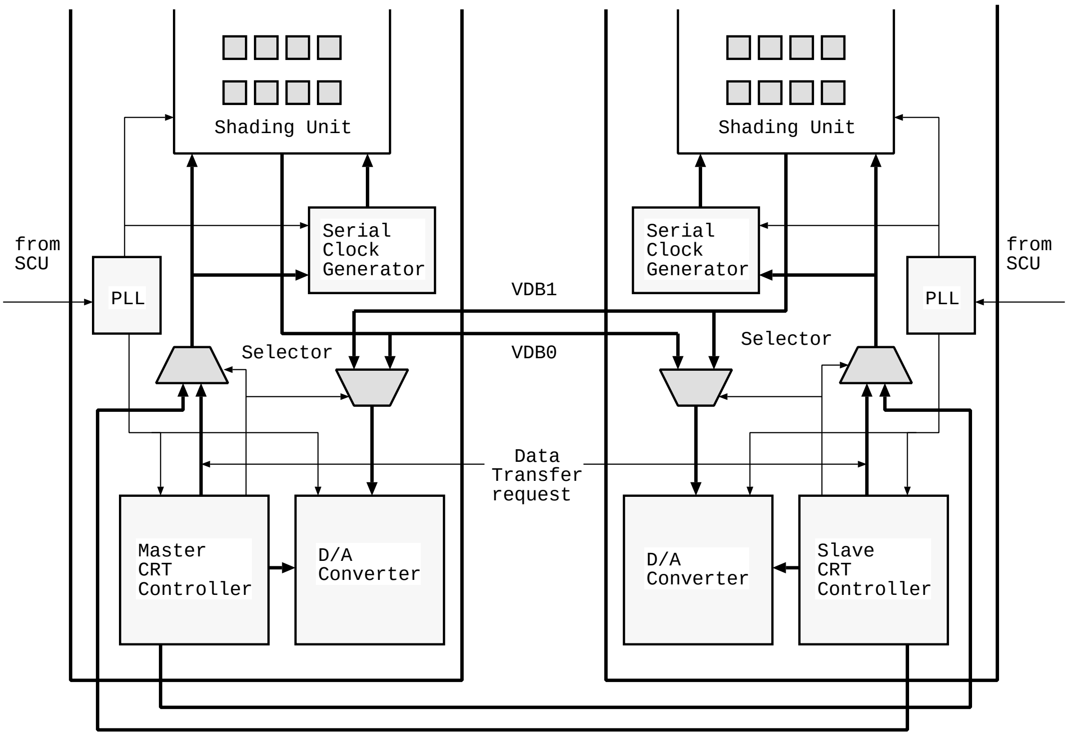


図 21: VSG の全体構成

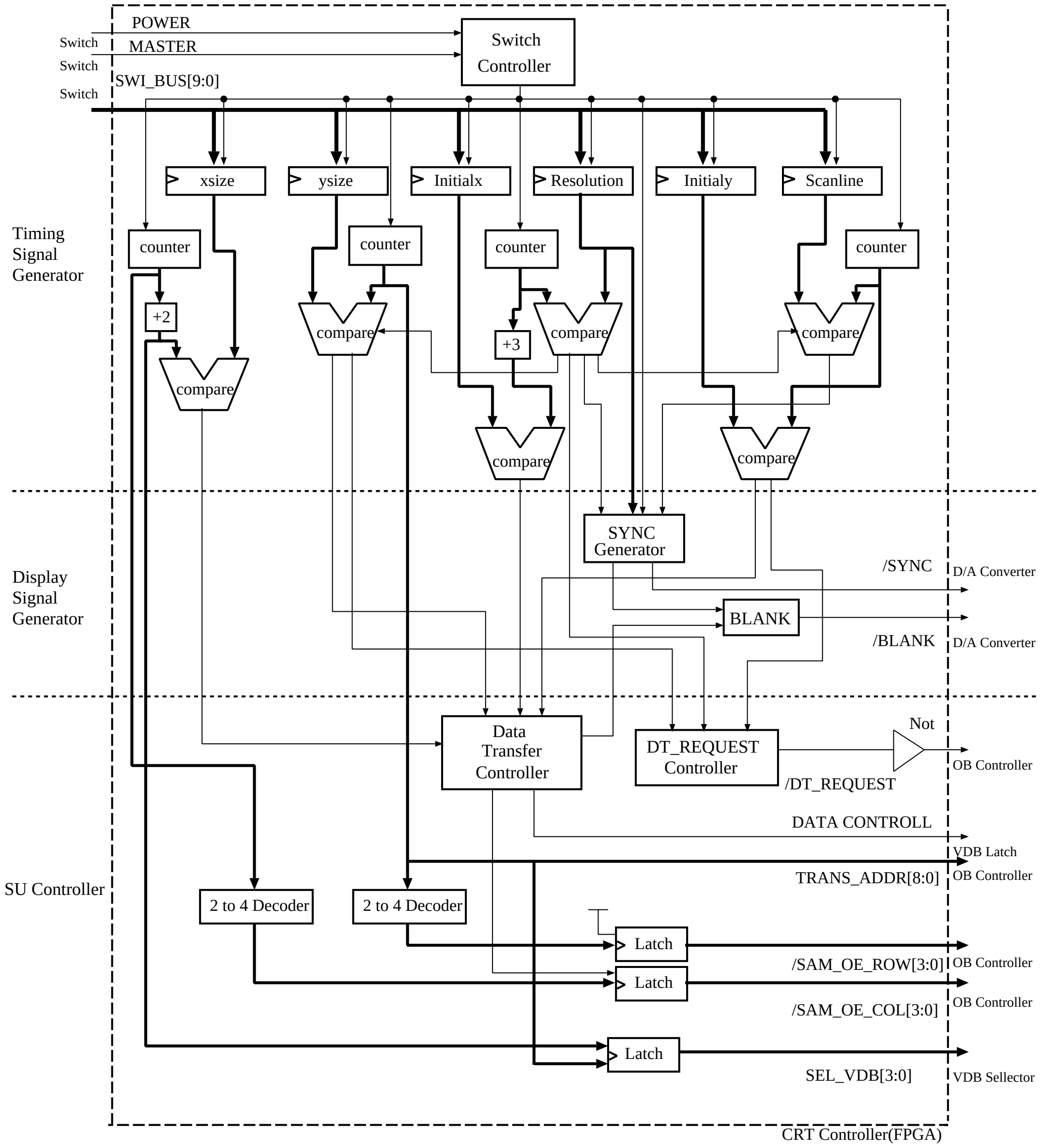


図 22: CRT Controller の全体構成

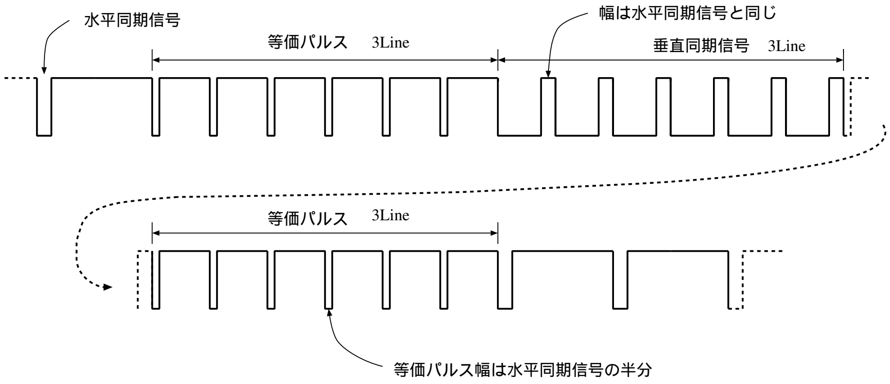


図 24: 垂直同期信号付近の信号

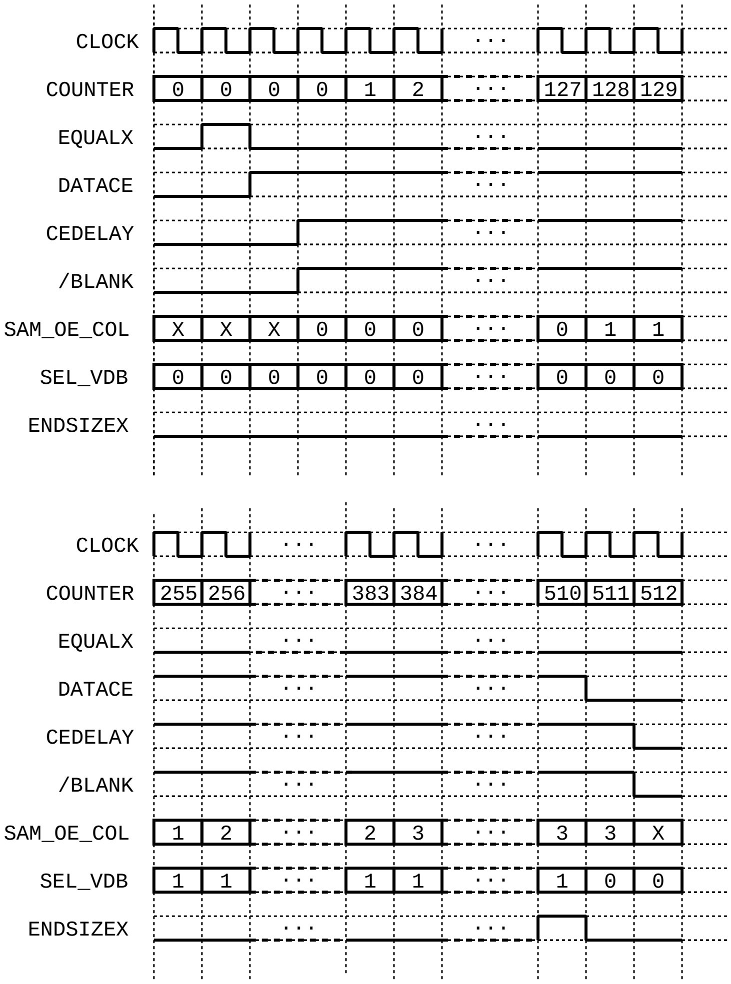


図 26: RGB データ転送のタイミング (表示画面が 512^2 の場合)