

ホログラフィ専用プロセッサの 開発による次世代映像システム

2021年2月

千葉大学大学院融合理工学府
基幹工学専攻電気電子工学コース

山本 洋太

(千葉大学審査学位論文)

ホログラフィ専用プロセッサの 開発による次世代映像システム

2021年2月

千葉大学大学院融合理工学府
基幹工学専攻電気電子工学コース

山本 洋太

概要

ホログラフィは、光の干渉・回折を利用し、3次元情報の記録・再生を行える技術である。計算機によって生成した Computer-Generated Holograms を利用した電子ホログラフィは、人間の立体知覚要因のすべてを満たせる3次元ディスプレイを実現可能である。光学的に撮影したホログラムを計算機で解析するデジタルホログラフィでは、ハイスピードカメラを利用した撮影により、瞬間的な3次元物体の観測を可能にする。いずれも計算機を利用するが、必要とされる計算量の大きさが実用化への問題になっている。本論文では、計算の超高速化を実現することにより、電子・デジタルホログラフィ実現への指針を示すことを目的とした。回路を再構成可能な Field Programmable Gate Array (FPGA) を用いて、チップレベルでの専用計算回路の並列数を向上することで、高速な計算機システムの開発を行った。

先行研究では、電子ホログラフィ専用計算機として、FPGA チップを8個搭載したボードを8つ利用し、計64個のFPGAチップを利用したクラスタシステムの開発が行われた。専用計算回路を実装した複数のFPGAチップにより、電子ホログラフィ計算が高速化されることが実証された。本研究では、チップ単体での高速化に注目し、チップレベルでのさらなる並列度向上が可能であることを検証した。開発では2種類のFPGAを利用した専用計算機開発を行った。一つは、FPGAと組み込みCPUをワンチップに搭載した System on a Chip (SoC) を用いて、将来的に Head Mounted Display (HMD) に組み込んで利用可能なチップレベルで小型なシステムの検討を行った。もう一つは、最新のFPGAチップを搭載したFPGAボードを利用することで、さらなる集積度向上が可能であることを検証した。開発した専用計算機システムは、CPU (PC) と比べ、最大230倍の高速化を達成した。専用計算機の一つである Graphics Processing Unit (GPU) と比べても7倍の高速化を実現した。

デジタルホログラフィ専用計算機では、ハイスピードカメラを用いた小さい画素数 (128×128画素) 向けの計算機を開発した。ハイスピードカメラでは、一般的に撮影フレームレートが増える程、画素数は小さくなる。画素数が小さい場合は、並列実行に必要なデータコピーなどの準備により、通常の計算機では並列化による計算効率が下がり、高速化が期待できない。一方で、専用計算機では、並列実行時に事前準備は必要なく、用意された計算回路によって計算が実行されるため、高速化が期待できる。デジタルホログラフィ専用計算機が通常のシステムでは高速化が期待できない画素数であっても高速に計算可能であることを検証した。

本論文の専用計算機を用いることで、電子ホログラフィでは、観察者に負担をかけないHMDを実現可能である。デジタルホログラフィでは、顕微鏡サイズで持ち運び可能なデジタルホログラフィ顕微鏡が実現可能となる。

Next-generation video system by special-purpose processor for holograpy

Yota Yamamoto

Abstract: Holography is a technique for recording and reproducing three-dimensional (3D) information using light interference and diffraction. Electroholography using computer-generated holograms can produce natural 3D scenes and has gained recognition as an ideal 3D technology. Digital holography, a computational analysis of optically captured holograms, enables instantaneous observation of 3D objects using a high-speed camera. Both of them use computers, but the required computation amount is a problem for practical use. The purpose of this dissertation is to provide a guideline for the realization of electroholography and digital holography by achieving ultra-high-speed computation. I have developed high-speed computer systems using Field Programmable Gate Arrays (FPGAs) that can reconfigure circuits.

A cluster system with 64 FPGA chips was developed in the previous work using eight boards with eight FPGA chips as a special-purpose computer for electroholography. And, it was demonstrated that multiple FPGA chips with special-purpose circuits could accelerate electroholography calculations. In this study, I focus on speeding up the single chip and verify that it is possible to improve the parallelism at the chip level. I have developed two types of FPGA-based computers. A system on a Chip (SoC) with an FPGA and an embedded CPU on a single chip was used to study a compact chip-level system that could be used for future head mounted displays (HMDs). The second is to verify that further integration can be achieved by using an FPGA board with the latest FPGA chips. The special-purpose computer system is up to 230 times faster than the CPU (PC). The speed is seven times faster than that of a Graphics Processing Unit (GPU).

In digital holography, I developed a special-purpose computer for captured holograms with a small size of 128×128 pixels using a high-speed camera. In a high-speed camera, the number of pixels generally decreases as the frame rate increases. When the number of pixels is small, parallelization efficiency is reduced on ordinary computers due to the preparation of data copying and other tasks required for parallel execution, and thus speeding up the process cannot be expected. On the other

hand, the special-purpose computer can accelerate the reconstruction calculations in parallel without any prior preparations. I verified that the special-purpose computer could compute fast even with the number of pixels that cannot be accelerated by conventional systems.

By using the special-purpose computer for electroholography, it is possible to realize an HMD without any burden to the observer. The special-purpose computer for digital holography makes it possible to realize a portable digital holographic microscope.

目次

序論	1
第1章 ホログラフィ	6
1.1 ホログラフィの原理	6
1.2 コンピュータホログラフィ	8
1.3 電子ホログラフィの原理	9
1.3.1 点光源モデルによる CGH 計算	10
1.3.2 振幅型ホログラムと位相型ホログラム	12
1.4 電子ホログラフィ計算の高速化手法	13
1.4.1 フレネル近似を用いた計算	13
1.4.2 漸化式法を用いた計算	14
1.5 デジタルホログラフィの原理	16
1.5.1 フレネル-キルヒホッフの回折積分公式	17
1.6 デジタルホログラフィ計算の高速化手法	18
1.6.1 フレネル近似を用いた計算	18
1.6.2 畳み込み回折積分法を用いた計算	19
第2章 VC707を用いたデジタルホログラフィ専用計算機	21
2.1 デジタルホログラフィ専用計算機 FFT-HORN の開発	21
2.2 デジタルホログラフィ専用計算機の構成	22
2.3 デジタルホログラフィ専用計算回路の実装	24
2.3.1 2次元畳み込み計算回路の実装	24
2.3.2 G 計算ユニットの実装	28
2.4 リソース使用率	28
2.5 計算性能比較	29
2.6 再生像評価	31
2.7 小括	32
第3章 HORN-8を用いた大規模な電子ホログラフィの再生	34
3.1 電子ホログラフィ専用計算機 HORN の開発	34
3.2 HORN-8 システムの開発	35

3.3	時空間分割法	36
3.4	HORN-8 の性能	37
3.5	光学系	39
3.6	再生像	39
3.7	小括	40
第4章	Zynq UltraScale+ MPSoC を用いた電子ホログラフィ専用計算機	41
4.1	Zynq UltraScale+ MPSoC について	41
4.2	組み込みシステム向け電子ホログラフィ専用計算機の構成	42
4.2.1	ターゲットボード	42
4.3	組み込み CPU・FPGA 間通信	45
4.3.1	AXI プロトコルの基礎	45
4.3.2	高速な AXI 通信方法	47
4.3.3	AXI 通信回路の構成	49
4.3.4	AXI Lite 通信回路の実装	50
4.3.5	AXI-Full 通信回路の実装	54
4.4	振幅型電子ホログラフィ専用計算回路	58
4.4.1	振幅型の計算精度設定	59
4.4.2	振幅型の回路構成	63
4.4.3	漸化式計算ユニットの実装	65
4.4.4	初期位相計算ユニットの実装	66
4.4.5	加算位相計算ユニットの実装	67
4.4.6	光強度計算ユニットの実装	67
4.4.7	リソース使用率	68
4.4.8	計算性能評価	68
4.4.9	光学再生	70
4.5	位相型電子ホログラフィ専用計算回路	72
4.5.1	位相型の計算精度設定	72
4.5.2	位相型の回路構成	75
4.5.3	漸化式計算ユニットの実装	76
4.5.4	初期位相計算ユニットの実装	78
4.5.5	加算位相計算ユニットの実装	79
4.5.6	複素振幅計算ユニットの実装	79
4.5.7	リソース使用率	80
4.5.8	計算性能評価	80
4.5.9	光学再生	82
4.6	電子ホログラフィを用いたインタラクティブシステム	84

4.6.1	Leap Motion について	84
4.6.2	Leap Motion を用いたインタラクティブシステム	84
4.6.3	インタラクティブシステムの処理時間	86
4.7	小括	87
第 5 章	Alveo U250 を用いた電子ホログラフィ専用計算機	89
5.1	電子ホログラフィ専用計算機の構成	89
5.1.1	専用計算システムの構成	90
5.2	PC・FPGA ボード間通信	91
5.2.1	通信回路の実装	91
5.2.2	write responder の実装	93
5.2.3	read responder の実装	95
5.2.4	アドレスマップ	97
5.3	振幅型電子ホログラフィ専用計算回路	98
5.3.1	漸化式計算ユニットの実装	100
5.3.2	初期位相計算ユニットの実装	101
5.3.3	加算位相計算ユニットの実装	102
5.3.4	光強度計算ユニットの実装	103
5.3.5	リソース使用率	104
5.3.6	計算性能評価	104
5.4	位相型電子ホログラフィ専用計算回路	106
5.4.1	漸化式計算ユニットの実装	107
5.4.2	複素振幅計算ユニットの実装	109
5.4.3	リソース使用率	109
5.4.4	計算性能評価	110
5.5	小括	111
結論		113
参考文献		121
謝辞		122
業績リスト		123

目次

1	両眼視差方式の原理と輻輳調節矛盾.	2
2	ホログラフィの再生イメージ.	3
1.1	ホログラフィの原理.	7
1.2	コンピュータホログラフィ.	8
1.3	点光源が1点とホログラムの位置関係.	10
1.4	複数の点群とホログラムの位置関係.	12
1.5	点光源 (球面波) の半径 z_0 の波面.	13
1.6	漸化式法の適用イメージ.	16
1.7	フレネル-キルヒホッフの回折積分公式による再生像計算.	17
2.1	Virtex-7 FPGA VC707 評価キットの外観. 黒色のファンの下に FPGA チップが搭載されている.	22
2.2	デジタルホログラフィ専用計算機システム全体のブロック図.	23
2.3	デジタルホログラフィ専用計算回路のブロック図 (VC707 内部).	25
2.4	2次元畳み込み計算ユニットのブロック図.	26
2.5	G unit のブロック図.	28
2.6	デジタルホログラフィ用光学系の様子. HM は Half Mirror を表す.	31
2.7	CPU と FPGA (VC707) による再生像計算の比較.	31
3.1	HORN-8 システムの外観: (a) 1 台の PC に 2 台の HORN-8 ボードを搭 載している様子 (b) HORN-8 クラスタ (4 台の PC に 8 台の HORN-8 ボードを搭載している様子).	36
3.2	HORN-8 システムを用いた時空間分割法適用イメージ.	37
3.3	HORN-8 システムにおける点群数と計算時間の関係.	38
3.4	時空間分割法を用いた再生光学系の様子.	39
3.5	計算速度による再生像の違い.	40
4.1	専用計算システム内部の役割分担.	42
4.2	Zynq UltraScale+ MPSoC ZCU102 評価キットの外観. 黒色のファン の下に FPGA チップが搭載されている.	44

4.3	AXI の通信モード (1 Channel).	46
4.4	AXI の通信波形 (1 Channel).	46
4.5	AXI の通信モード図 (全体).	47
4.6	バースト転送のイメージ.	48
4.7	バースト転送とオーバーラップ転送を利用しない場合の転送イメージ.	48
4.8	オーバーラップ転送のイメージ.	48
4.9	電子ホログラフィ専用計算機向け AXI 通信回路の様子.	49
4.10	AXI Lite 通信回路のブロック図.	50
4.11	write responder の状態遷移図.	51
4.12	read responder の状態遷移図.	51
4.13	AXI-Full 通信回路のブロック図.	55
4.14	バッファ FIFO の入出力端子の様子.	55
4.15	write controller の状態遷移図.	57
4.16	read controller の状態遷移図.	58
4.17	電子ホログラフィ計算における整数値の固定小数点数表現.	60
4.18	固定小数点数表現.	61
4.19	電子ホログラフィ計算における小数を含む固定小数点数表現.	61
4.20	再生像の PSNR と CGH の MSE 比較 (振幅型).	62
4.21	再生像の SSIM と CGH の MSE 比較 (振幅型).	62
4.22	再生シミュレーション像の比較 (振幅型 CGH).	63
4.23	振幅型電子ホログラフィ専用計算回路のブロック図.	63
4.24	漸化式計算部のブロック図.	65
4.25	初期位相計算部のブロック図.	66
4.26	加算位相計算部のブロック図.	67
4.27	光強度計算部のブロック図.	68
4.28	計算時間の比較.	69
4.29	点光源データの様子 (キューブ).	70
4.30	光学再生した動画の一部 (キューブ).	71
4.31	点光源データの様子 (恐竜の骨格).	71
4.32	光学再生した動画の一部 (恐竜の骨格).	71
4.33	点光源データの様子 (チェス盤).	72
4.34	光学再生した動画の一部 (チェス盤).	72
4.35	正規化の例.	73
4.36	再生像の PSNR と CGH の MSE 比較 (位相型).	74
4.37	再生像の SSIM と CGH の MSE 比較 (位相型).	74
4.38	再生シミュレーション像の比較 (位相型 CGH).	75

4.39	位相型電子ホログラフィ専用計算回路のブロック図.	75
4.40	漸化式計算部のブロック図.	77
4.41	初期位相計算部のブロック図.	78
4.42	加算位相計算部のブロック図.	79
4.43	複素振幅分布計算部のブロック図.	80
4.44	計算時間の比較.	81
4.45	点光源データの様子 (キューブ).	82
4.46	光学再生した動画の一部 (キューブ).	82
4.47	点光源データの様子 (恐竜の骨格).	83
4.48	光学再生した動画の一部 (恐竜の骨格).	83
4.49	点光源データの様子 (チェス盤).	83
4.50	光学再生した動画の一部 (チェス盤).	83
4.51	Leap Motion の外観.	84
4.52	電子ホログラフィ専用計算機を利用したインタラクティブシステム の構成.	85
4.53	インタラクティブ処理システム (専用計算機システム内).	86
4.54	インタラクティブに3次元映像を操作している様子. 指の動きに合わ せて再生像が移動している.	87
5.1	Alveo U250 データセンターアクセラレータカードの外観. 外装に覆わ れており, FPGA チップは確認できない.	90
5.2	専用計算システム全体の構成.	91
5.3	AXI 通信回路の様子.	92
5.4	write responder (アドレス転送用) の遷移図.	93
5.5	write responder (データ転送用) の遷移図.	94
5.6	read responder (アドレス転送用) の遷移図.	95
5.7	read responder (データ転送用) の遷移図.	96
5.8	振幅型電子ホログラフィ専用計算回路のブロック図.	99
5.9	漸化式計算部のブロック図. SEL は Selector を表す.	101
5.10	初期位相計算部のブロック図.	102
5.11	加算位相計算部のブロック図.	103
5.12	光強度計算部のブロック図.	103
5.13	計算時間の比較.	105
5.14	位相型電子ホログラフィ専用計算回路のブロック図.	106
5.15	漸化式計算部のブロック図.	108
5.16	複素振幅分布計算部のブロック図.	109
5.17	計算時間の比較.	110

表目次

2.1	VC707 の論理回路リソース.	23
2.2	デジタルホログラフィ専用計算機のリソース利用率.	29
2.3	デジタルホログラフィ再生計算時間の比較.	30
2.4	ホログラムの再生計算条件.	32
3.1	CPU, GPU, HORN-8 ボード単体, HORN-8 クラスタシステムにお ける CGH 計算時間の比較.	38
4.1	ZCU102 の論理回路リソース.	44
4.2	ZCU102 の CPU 環境.	45
4.3	アドレスマップ (ZCU102).	54
4.4	振幅型 CGH 専用計算機のリソース使用率.	68
4.5	振幅型 CGH 計算時間の比較.	69
4.6	位相型 CGH 専用計算機のリソース使用率.	80
4.7	位相型 CGH 計算時間の比較.	81
5.1	U250 の論理回路リソース.	90
5.2	アドレスマップ (U250).	98
5.3	振幅型 CGH 専用計算機のリソース使用率.	104
5.4	振幅型 CGH 計算時間の比較.	105
5.5	位相型 CGH 専用計算機のリソース使用率.	110
5.6	位相型 CGH 計算時間の比較.	111

序論

ホログラフィは、光の干渉・回折を利用し、3次元情報を記録・再生する技術である。1948年にD.Gaborによって顕微鏡の精度向上のため提案された手法 [1] を元とする。ホログラフィでは、3次元物体からの光の波面を写真乾板などの2次元媒体 (ホログラム) へ記録することで、3次元的な映像再生を可能とする。一般的な写真 (フォトグラフィ) が光波の振幅のみを記録する技術であるのに対して、ホログラフィは光波の振幅と位相の両方を記録する。写真上には光の強弱のみ保存されるため、観察視点によらず、記録したときの光の強弱のみ再現され、2次元的な映像となる。一方、ホログラフィは光の振幅と位相を保存するため、実際に3次元物体を観察しているときと同じ波面が再現され、3次元的な映像となる。

人間は奥行き手がかりと呼ばれる様々な網膜像情報を利用し、奥行きを知覚している。本来は2次元的な広がりしかない網膜像であるが、そこに含まれる情報から脳内で3次元情報を再構築し、認識している。奥行き手がかりは、眼球運動性のもの、両眼性のもの、そして単眼性のものに分けられる [2]。眼球運動性のものは、目のピント調節や輻輳と呼ばれる目の傾き具合により知覚される奥行き要因である。両眼性のものは、両眼の見え方の違い (両眼視差) により知覚される奥行き要因である。単眼性のものは、実際に奥行きがない2次元的な絵であっても、手前のものは大きく鮮明に見え、遠くのは小さくぼやけて見えるといった経験に基づく視覚要因である。

立体視が可能な装置の一つに Head Mounted Display (HMD) がある。各社からリリースされ、エンターテインメントや教育、医療など、様々な分野で活用され始めている。HMDでは、両眼視差方式と呼ばれる技術を利用している。2つの映像 (ディスプレイ) を左右に並べ、視差をつけた映像をそれぞれに提示することで観察者に立体映像を知覚させている。システム開発が比較的容易であり、広く普及している。ただし、図1に示すとおり、輻輳と目のピント調節により知覚される位置がずれてしまう問題点がある。この問題は特に輻輳調節矛盾と呼ばれ、長時間の利用により吐き気や眠気といったいわゆる3D酔いが発生する原因とされる [3,4]。また、左右に映像を並べるという構成のため、眼球は常に輻輳がついた状態 (目を傾けた状態) が続く。未成熟の子供が利用することで視覚障害が発生する可能性が指摘されている [5]。

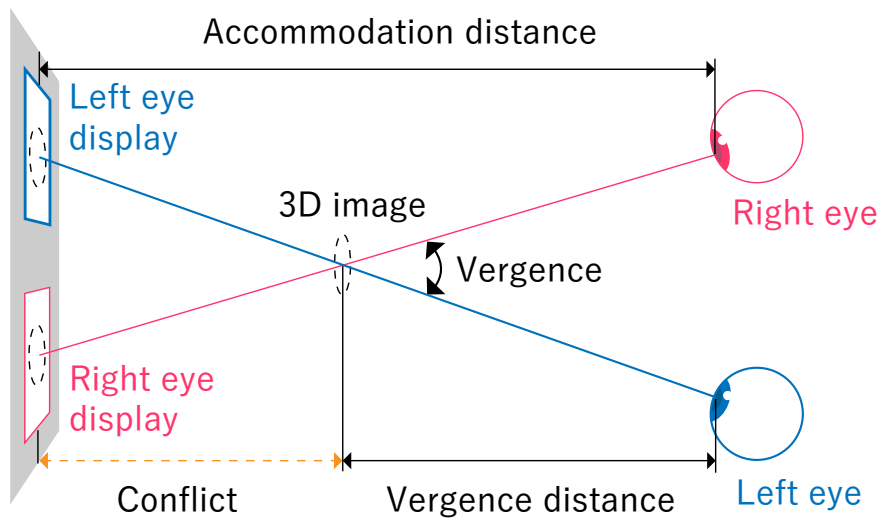


図 1: 両眼視差方式の原理と輻輳調節矛盾.

近年では、ホログラムといえば3次元映像を示す代名詞として多くの人に認知されている。ホロ、ホログラム、ホログラフィックなどの言葉冠した製品が多くリリースされている。それらは3次元ディスプレイを想起させるものであるが、実際に理想的な3次元映像を表示できているものは多くない。現在入手可能な立体ディスプレイの多くは必要な奥行き手がかりのすべてを完全に再現することができていない。輻輳調節矛盾といった問題をはじめとして、知覚(脳)へ何らかの負担を与える。一方で、ホログラフィでは、図2に示すようにすべての奥行き手がかりを再現可能である。輻輳調節矛盾が起こることもなく、観察者に負担をかけない理想的な3次元映像を提示可能である。

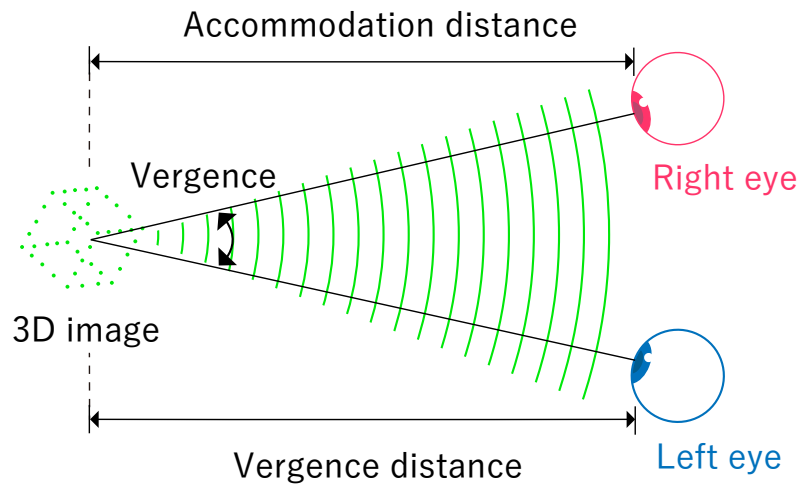


図 2: ホログラフィの再生イメージ.

従来の写真乾板などの2次元媒体を利用していたホログラフィをアナログホログラフィと呼ぶのに対し、電子的に実現されたホログラフィを特にコンピュータホログラフィと呼ぶ。コンピュータホログラフィは、電子ホログラフィ [6] とデジタルホログラフィ [7] に分けられる。計算機によって生成した Computer-Generated Holograms (CGHs) を利用した電子ホログラフィは、人間の立体知覚要因のすべてを満たせる3次元ディスプレイを実現することが可能である [8]。究極の3次元映像提示手法とも呼ばれ、注目を集めている。電子素子を利用し撮影したホログラムを計算機で解析するデジタルホログラフィでは、ハイスピードカメラを利用した撮影により、瞬間的な3次元物体の観測を可能にする [9–13]。位相を記録することにより、従来の撮影手法では記録が難しい流体や細胞などの透明物体を記録することも可能である [14]。次世代の計測技術として注目されている。

コンピュータホログラフィでは計算機を利用するが、必要とされる計算量の大きさが実用化への問題になっている [15–17]。先行研究では計算の高速化手法 [18–26] の研究や Graphics Processing Unit (GPU) [27–30] や Application Specific Integrated Circuits (ASICs) [31], Field-Programmable Gate Array (FPGA) [32–37] といったハードウェアを計算アクセラレータとして利用する研究が行われた。ハードウェアを用いた高速化は非常に効果的であり、アルゴリズムのみのソフトウェア的な計算高速化が数十倍であるのに対し、ハードウェアを用いた計算は数百倍を達成している [35]。

電子ホログラフィ専用計算機として1992年から開発が行われている専用計算機に Holographic Reconstruction (HORN) がある [38, 39]。試作機である HORN-1 は、汎用 Integrated Circuit (IC) を使い手配線によって計算回路の設計・実装が行われ、

CGH 計算における専用計算機の有効性が示された [38, 39]. その後, HORN-3 [40] より, FPGA を用いた計算回路の実装が行われている. 最新の電子ホログラフィ向け専用計算機である HORN-8 システムは, 独自に開発を行った専用基板ボードである HORN-8 ボードを 8 台用いたクラスタシステムである. HORN-8 ボード 1 台に FPGA チップを 8 個搭載し, 8 台のボードを利用したクラスタシステムにより 64 個の FPGA チップによる並列計算を可能とする [35, 36]. 一般的な計算機は, 並列化時にデータのコピーや準備が必要なため, 並列数と同じだけの高速化が得られないことが知られている. 一方で, HORN-8 システムは, 計算コアが増加するに従って計算が高速化するというホログラフィ専用計算機での有効性が示された [35].

デジタルホログラフィ向け専用計算機として開発された Fast Fourier Transform Holographic Reconstruction (FFT-HORN) は, 流体の可視化を目的とした Digital Holographic Particle Tracking Velocimetry (DHPTV) 向けに $1,024 \times 1,024$ 画素を超えるホログラムが計算可能なものが開発された [16, 17]. デジタルホログラフィ計算では計算に必要な Fast Fourier Transform (FFT) 計算にかかる時間が支配的である. FFT 計算をハードウェアで計算することにより, 効率的な計算が可能であることが示された [16, 17].

本研究では, 計算の超高速化を実現することにより, 電子・デジタルホログラフィ実用への指針を示すことを目的とした. 先行研究と同じく, 回路を再構成可能な FPGA を用いて高速な計算機システムの開発を行った. 電子ホログラフィ専用計算機では, 特に専用計算回路の集積度を向上し, チップレベルでの並列数を向上した. 実際の開発では, 2 種類の FPGA を利用した開発を行った. 一つは, FPGA と組み込み CPU をワンチップに搭載した System-on-a-chip (SoC) を用いて, 将来的に Head Mounted Display (HMD) に組み込んで利用可能なチップレベルで小型なシステムの検討を行った [41, 42]. もう一つは, 最新の FPGA チップを搭載した FPGA ボードを利用することで, さらなる集積度向上が可能であることを検証した. デジタルホログラフィ専用計算機では, ハイスピードカメラを用いた小さい画素数 (128×128 画素) 向けの計算機を開発した. ハイスピードカメラでは, 一般的に撮影フレームレートが増える程, 画素数は小さくなる. 画素数が小さい場合は, 並列実行に必要なデータコピーなどの準備により, 通常の計算機では並列化による計算効率ที่下がり, 高速化が期待できない. 一方で, 専用計算機では並列実行時に事前準備は必要なく, 用意された計算回路によって直ちに並列計算が実行されるため, 高速化が期待できる. デジタルホログラフィ専用計算機が通常のシステムでは高速化が期待できない画素数であっても高速に計算可能であること検証した [43].

1 章では, 本研究の基礎となるホログラフィの原理を示した. 光の干渉と回折を利用するホログラフィの原理を説明し, 電子的に実現されたコンピュータホログラフィの計算手法とその高速計算手法について紹介した. 2 章では, デジタルホログラ

フィ向け専用計算機の開発について示した。新たにハイスピードカメラを用いた小さい画素数 (128 × 128 画素) 向けの計算機の開発を行い，その実装とパフォーマンスについて示した。3章では，最新の HORN-8 システムを用いた大規模な電子ホログラフィ再生手法について示した。計算の高速化が実用上だけではなく，より高精細な3次元映像を提示するためにも重要であることを示した。4章では，FPGA と組み込み CPU をワンチップに搭載した SoC を用いた電子ホログラフィ専用計算機開発について示した。SoC を利用したシステムの実装とパフォーマンスについて紹介し，小型で高性能なシステムが構成可能なことを示した。また，ハンドモーショントラッカーを利用したインタラクティブシステムの開発を行い，HMD などのアプリケーションで有効に利用可能であることを示した。5章では，最新の大規模 FPGA チップを利用した電子ホログラフィ専用計算機開発について示した。大規模 FPGA 向けに設計し直したシステムの実装とそのパフォーマンスについて紹介し，さらなる高速化が達成可能であることを示した。

第1章 ホログラフィ

ホログラフィは、光の干渉・回折を利用し、3次元情報を記録・再生する技術である。ホログラフィでは、3次元的な光の波面が保存されたホログラムが重要な役割を果たす。近年では、ホログラムといえば立体映像を示す代名詞として認知されている。しかし、もともとは立体映像のための技術ではなく、顕微鏡の精度を上げるために開発された技術であった [1]。

当時、顕微鏡の世界は光学を利用した光学顕微鏡から、量子力学を利用した電子顕微鏡の世界となり、光学顕微鏡では観察できなかったウィルスなどの姿が観察されていた。しかし、電子顕微鏡には光学顕微鏡とは異なり、凹レンズが使えないという問題点があった。この問題点は、レンズにより生じるボケや歪みを消すことができないという問題となって現れていた。その解決策としてD.Gaborによって提案された技術こそがホログラフィである。物を見ることは、物理的にはものから発せられる散乱光をレンズ系で結像させることである。ホログラフィではこの過程を2段階に分ける。最初に電子線の散乱波を収差も含めてフィルムに記録する。次に、これに光を当てて結像させる。こうすることにより、光学手法に持ち込むことが可能であり、凹レンズが使える、収差を消すことが可能である。「ホロ」とはギリシャ語で「すべて」を意味し、「グラム」は記録を意味する。その後、ホログラムに関する技術全般をホログラフィと呼ぶようになったとされる [44]。

本来のホログラムの意味を超えて、ホロ、ホログラム、ホログラフィックなどの言葉を冠した製品が多くリリースされている。それらは3次元ディスプレイを想起させるものであるが、D.Gaborによるホログラフィを特徴づける光の波面を再現した理想的な3次元映像を提示できているものは多くない。本研究では、ホログラフィの歴史に沿い、光の波面が保存された干渉縞をホログラム、ホログラムを用いた技術全般をホログラフィと呼ぶ。ホログラムから再生される立体映像は再生像と呼ぶ。

1.1 ホログラフィの原理

ホログラフィは3次元物体からの光の波面を写真乾板などの2次元媒体 (ホログラム) へ記録する技術である。一般的な写真が光波の振幅のみを記録するのに対して、ホログラフィは光波の振幅と位相の両方を記録する。写真上には光の強弱のみ

保存されるため、観察視点によらず記録したときの光強度のみ再現され、2次元的な映像となる。一方、ホログラフィは光の振幅と位相を保存するため、実際に3次元物体を観察しているときと同じ波面が再現され、3次元的な映像となる。

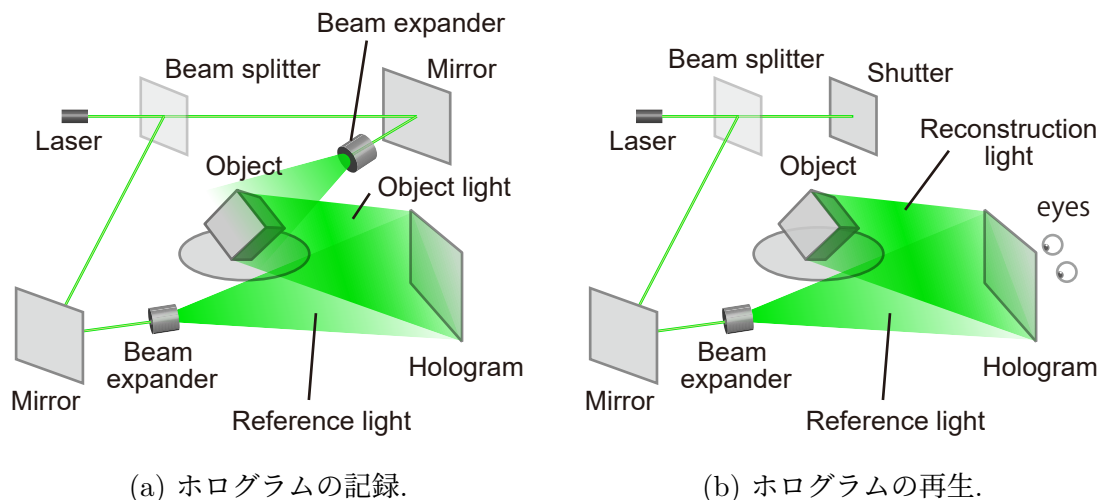


図 1.1: ホログラフィの原理.

ホログラフィの原理は記録と再生プロセスに分けられる (図 1.1). ホログラフィにおいて、位相の揃った光源によって記録する。位相が揃った状態をコヒーレンスといい、干渉縞を形成出来ることから可干渉性と呼ばれる。代表的なコヒーレント光にレーザーがあり、ホログラフィにおいてよく用いられる。

ホログラフィの記録プロセスでは、コヒーレント光をビームスプリッターにより二つに分割する。一方を記録したい物体に当て、物体光とする。もう一方をそのまま参照光とする。そして、物体光と参照光を記録素材上で重ね合わせる。記録素材上では光の位相のずれにより、干渉縞 (ホログラム) が出現し、記録される。このとき、ホログラム面 (x_a, y_a) 上での物体からの反射光 (物体光) を $O(x_a, y_a)$ 、光源からの光 (参照光) を $R(x_a, y_a)$ とおくと、物体光と参照光を足し合わせることで、ホログラムの干渉縞である光強度分布 $I(x_a, y_a)$ は式 (1.1) で計算される。ここで、 $*$ は複素共役を表す。

$$\begin{aligned}
 I(x_a, y_a) &= |O(x_a, y_a) + R(x_a, y_a)|^2 \\
 &= |O(x_a, y_a)|^2 + |R(x_a, y_a)|^2 + O(x_a, y_a)R^*(x_a, y_a) + O^*(x_a, y_a)R(x_a, y_a)
 \end{aligned}
 \tag{1.1}$$

式 (1.1) において、第1項は物体光 $O(x_a, y_a)$ の自己相関、第2項は参照光 $R(x_a, y_a)$ の自己相関、第3項は物体光 $O(x_a, y_a)$ に比例する項、第4項は物体光の共役像 $O^*(x_a, y_a)$ に比例する項である。

再生プロセスでは、記録時と同じ波長を持つ光を干渉縞 (ホログラム) が記録された記録素材に照射することにより、ホログラム上で光が回折し、物体光が再現される。ホログラム $I(x_a, y_a)$ に参照光 $R(x_a, y_a)$ を照射するため、式 (1.1) より数学的には式 (1.2) で表される。

$$I(x_a, y_a) \times R(x_a, y_a) = R(x_a, y_a) \{ |O(x_a, y_a)|^2 + |R(x_a, y_a)|^2 \} + O(x_a, y_a) + O^*(x_a, y_a)R^2(x_a, y_a) \quad (1.2)$$

式 (1.2) において、右辺第 2 項が物体光そのものとなっていることがわかる。この回折光が物体光と同じ役割をすることで、3次元物体が存在するよう見える。

1.2 コンピュータホログラフィ

写真乾板などの 2次元媒体を利用した光学系によるホログラフィを特にアナログホログラフィと呼ぶ。アナログホログラフィを電子的に実現した技術にコンピュータホログラフィがある [44]。コンピュータホログラフィでは、アナログホログラフィにおける記録と再生プロセスそれぞれが大きな分野を形成している。記録に相当する技術を特に電子ホログラフィ (図 1.2 (a)) と呼び、再生に相当する技術をデジタルホログラフィ (図 1.2 (b)) と呼ぶ。

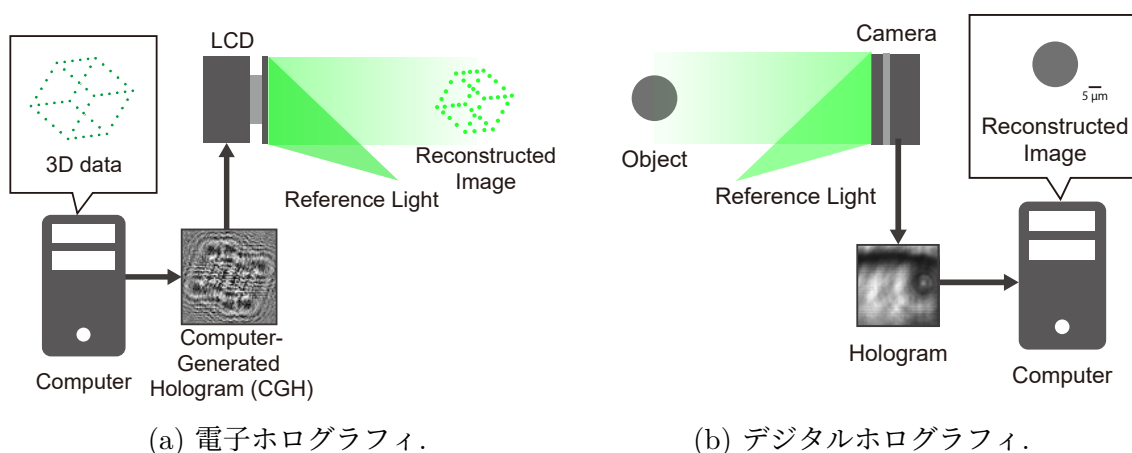


図 1.2: コンピュータホログラフィ.

図 1.2 (a) の電子ホログラフィは、コンピュータで計算によってホログラムを生成し、光学系を用いて 3次元像を投影する技術である。計算によって生成されたホログラムを特に計算機合成ホログラム Computer-Generated Hologram (CGH) と呼ぶ。CGH を使うことで、PC 上でモデリング可能であれば、実際に存在しない物体の投影が行える。CGH を Liquid Crystal Display (LCD) などの液晶ディスプレイに

表示し、画面表示を高速に切り替えることで動画の再生も可能である [6]。3次元テレビといった応用が研究されている。

図 1.2 (b) のデジタルホログラフィではレンズやカメラなどの光学系でホログラムを撮影し、コンピュータ内で計算によって3次元像を可視化する技術である。デジタルホログラフィでは、1枚のホログラムから任意の奥行き3次元情報を可視化可能である。また、位相を記録することにより、従来の撮影手法では記録が難しい流体や細胞などの透明物体現象を記録することも可能である [14]。3次元物体の測定や位置推定である Digital Holographic Particle Tracking Velocimetry (DHPTV) の研究が行われている [9–13]。ハイスピードカメラを用いることで、瞬間的な3次元情報解析への応用も期待されている [45]。

1.3 電子ホログラフィの原理

電子ホログラフィでは、計算によって生成する CGH を利用し、3次元映像の投影を可能にする。アナログホログラフィが実物体を利用し、その反射光 (物体光) と参照光からホログラムを得るのに対して、電子ホログラフィでは電子データ上の3次元情報に対して光波をシミュレートすることで CGH を生成する。

電子ホログラフィを利用したシステム開発は難しく、実用化し普及するまでにはまだしばらく時間を要すると考えられる。課題は大きく分けて4点ある。1つめに、表示用ディスプレイの高解像度化である。光の干渉・回折を利用するため、利用するディスプレイには数百 nm といった非常に細かな画素間隔を持ったディスプレイが要求される。2つめとして、ディスプレイの大画面化である。画素間隔が小さくなればなる程、同じ画素数ではディスプレイ自体のサイズは小さくなる。大画面化には膨大な画素数を表示可能なディスプレイが必要である。3つめとして、再生像の高画質化がある。これは、ディスプレイの高解像度・大画面化によっても達成可能である。一方で、原理的なところで参照光としてレーザ光源を利用するため、再生像にはスペックルノイズと呼ばれるノイズが発生するなどの問題点がある。再生像をより綺麗に表示する CGH 計算手法や光学系が求められている。高解像度・大画面化に関しては、1 μm としたディスプレイが開発されており [46,47]、8K4K としたディスプレイが開発される [48] など実用へ近づいている。また、高画質化に関しても様々な手法が提案されている [49–51]。一方で、4つめとして、CGH 計算の超高速化に関しては非常に難しい状態が続いている [15]。電子ホログラフィでは計算機を利用して CGH を生成するが、必要とされる計算量は非常に膨大である。3次元テレビといった利用法を考え、50万点で表現された3次元物体から6,000億画素の CGH を計算し、30 fps で表示することを考えたとき、ゼタスケール (10^{21} FLOPS: Floating-point Operations Per Second) の計算量が求められる。現状最も高性能な

スーパーコンピュータですらペタスケール (10^{16} FLOPS) であり、その乖離は大きい。年々その計算性能を増やしてきた計算機ではあるが、発案から 30 年近くたった現在でも実用には届いていない。

CGH の計算時間削減のため多くの研究が行われている。Look-up table 法を用いた研究 [18, 25], 差分を利用した研究 [19, 21], 波面記録面を利用した研究 [22, 23], スパース性を利用した研究 [26, 52], パッチモデルを利用した研究 [24], ポリゴンモデルを利用した研究 [53, 54], 光線波面変換を利用した研究 [55, 56], 光線法を利用した研究 [57, 58] などが行われている。中でも点光源モデルは、自発光する点光源によって 3 次元像を表現するモデルであり、計算が単純で並列化が容易な特徴を持つ。並列化が容易であるという特徴は、専用計算機実装時に高速化が期待できる。本研究は、専用計算機実装時の計算パイプラインの単純さと高速化への期待から点光源モデルを利用した。

1.3.1 点光源モデルによる CGH 計算

図 1.3 のように点光源が 1 点で平行光の参照光がホログラム面に対して垂直に入射するように設定する。点光源はホログラム中心から奥行き方向に z_0 だけ離れた位置におくと、点光源から発する光波は球面波として考えることができ、ホログラム上の座標 $(x_a, y_a, 0)$ において式 (1.3), 式 (1.4) で表される。ここで、 $k = 2\pi/\lambda$ は波数である。

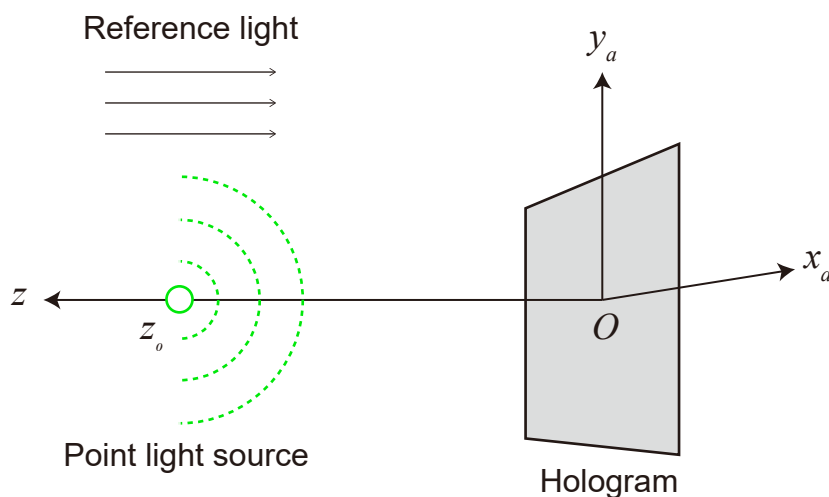


図 1.3: 点光源が 1 点とホログラムの位置関係。

$$O(x_a, y_a) = \frac{A}{r} \exp(i(kr + \phi)) \quad (1.3)$$

$$r = \sqrt{x_a^2 + y_a^2 + z_0^2} \quad (1.4)$$

計算を簡単にするため、定数である点光源の振幅 A と初期位相 ϕ はそれぞれ 1 と 0 に設定する。このとき、式 (1.3) は式 (1.5) で表される。

$$O(x_a, y_a) = \frac{1}{r} \exp(ikr) \quad (1.5)$$

参照光も同様に、定数である点光源の振幅 A_R と初期位相 ϕ_R はそれぞれ 1 と 0 に設定すると、式 (1.6) で表される。

$$\begin{aligned} R(x_a, y_a) &= A_R(x, y) \exp(i\phi_R(x_a, y_a)) \\ &= 1 \times \exp(i \cdot 0) = 1 \end{aligned} \quad (1.6)$$

このとき、物体光と参照光がホログラム面 (x_a, y_a) で重なり合うので、式 (1.1) より、ホログラム面上における光波の分布は式 (1.7) で求められる。

$$\begin{aligned} I(x_a, y_a) &= |O(x_a, y_a) + R(x_a, y_a)|^2 \\ &= \left| \frac{1}{r} \exp(ikr) + 1 \right|^2 \\ &= \frac{1}{r^2} + 1 + \frac{1}{r} \exp(ikr) + \frac{1}{r} \exp(-ikr) \end{aligned} \quad (1.7)$$

式 (1.7) の第 1 項と第 2 項はそれぞれ物体光と参照光の直接光であり、3次元再生に寄与しない。第 1 項と第 2 項を外し、式 (1.7) は式 (1.8) と表される。

$$\begin{aligned} I(x_a, y_a) &= \frac{1}{r} \exp(ikr) + \frac{1}{r} \exp(-ikr) \\ &= \frac{2}{r} \cos(kr) \end{aligned} \quad (1.8)$$

係数 2 は全体にかかるだけで本質的な意味合いは無い。従って、実際に計算するには、式 (1.8) より、式 (1.9) を求めればよい。

$$I(x, y) = \frac{1}{r} \cos(kr) \quad (1.9)$$

複数の点からなるホログラムは図 1.4 のように点光源ごとの光強度の重ね合わせで求めることができる。式 (1.9) より、点光源数を M 、点群座標を (x_j, y_j, z_j) 、CGH 上の座標を $(x_a, y_a, 0)$ と置くと、複数点からなる光強度はホログラム面 $I(x_a, y_a)$ において式 (1.10)、式 (1.11) で求められる。

$$I(x_a, y_a) = \sum_{j=0}^{M-1} \frac{1}{r_{aj}} \cos(kr_{aj}) \quad (1.10)$$

$$r_{aj} = \sqrt{(x_a - x_j)^2 + (y_a - y_j)^2 + (z_j)^2} \quad (1.11)$$

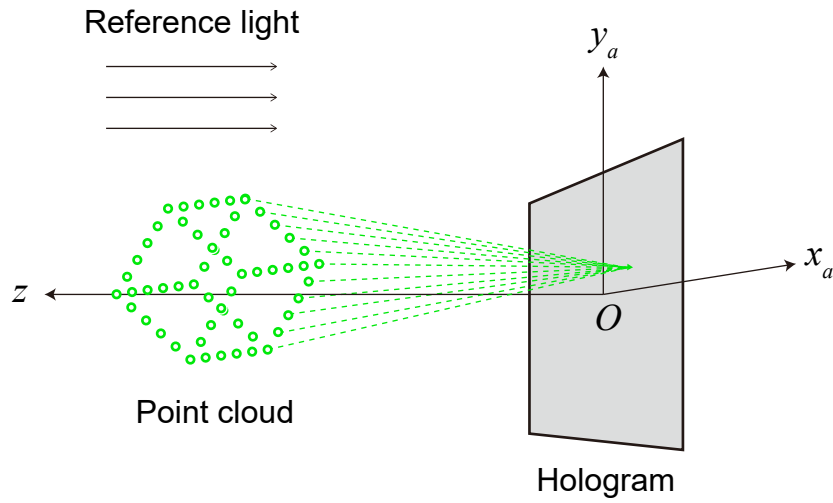


図 1.4: 複数の点群とホログラムの位置関係.

1.3.2 振幅型ホログラムと位相型ホログラム

式 (1.10) をホログラム面上で計算して記録したものを振幅ホログラムと呼ぶ. 式 (1.10) の結果は実数なので, そのまま振幅変調型 LCD に表示可能である. 一方で, 本来は必要無い共役光がノイズとなり, 画質を劣化させる要因となる. また, 干渉縞の濃淡で入射光の強度を弱めてしまうため, 光の利用効率が悪い [59]. 入射光に対し, 再生像に寄与する割合を表す回折効率, 振幅ホログラムでは 10%以下である [44]. 干渉縞の分布を位相変化で表すホログラムに位相ホログラムがある. 位相ホログラムは各画素で位相変調を行うことで干渉縞を表現するため, 入射光の振幅強度を弱めることがなく, 振幅型と比較して明るい再生像が得られる. 作り方によっては, 100%近い回折効率となる [44].

振幅型ホログラムは式 (1.2) の第 3 項と第 4 項を加算した. 位相型ホログラムは, 第 3 項のみを計算し, 位相情報をとり出す. 式 (1.2), 式 (1.10), 式 (1.11) より, 第 3 項は式 (1.12) で実部と虚部を分けて計算可能である.

$$\begin{aligned}
 O(x_a, y_a)R^*(x_a, y_a) &= \sum_{j=0}^{M-1} \frac{1}{r_{aj}} \exp(ikr_{aj}) \\
 &= \sum_{j=0}^{M-1} \frac{1}{r_{aj}} \cos(kr_{aj}) + i \sum_{j=0}^{M-1} \sin(kr_{aj}) \\
 &= \text{Re} + i\text{Im}
 \end{aligned} \tag{1.12}$$

位相ホログラムは, 式 (1.12) から式 (1.13) を用いて位相部分を抽出することで計算

する.

$$\phi(x_\alpha, y_\alpha) = \tan^{-1} \left(\frac{\text{Im}}{\text{Re}} \right) \quad (1.13)$$

電子ホログラフィにおいて、振幅型は位相型と比べ計算コストが小さく、高速に計算可能である。一方、位相型の計算コストは高いが、振幅型と比べ良質な再生像が得られる [59,60].

1.4 電子ホログラフィ計算の高速化手法

CGHを作成するためには、振幅型は式 (1.10)、位相型は式 (1.12) と式 (1.13) をホログラムの全画素について行う必要がある。総画素数を K 、点群数を M と置くと $O(KM)$ の計算量が必要である。また、計算負荷が高い三角関数演算や平方根演算が含まれるため、CGH作成にかかる計算負荷はさらに大きくなる。計算負荷削減のため、近似化手法や計算フローを工夫した計算が行われている。

1.4.1 フレネル近似を用いた計算

ホログラフィは物体光の波面を記録する技術である。点光源から発せられる波面は球面波となって進む。式 (1.3), 式 (1.4) では物体の表面 (点光源そのもの) を基準にして計算した。点光源の波面を記録するには、球波面を記録してもよい。図 1.5 のような半径 z_0 の球面波を記録を考える。

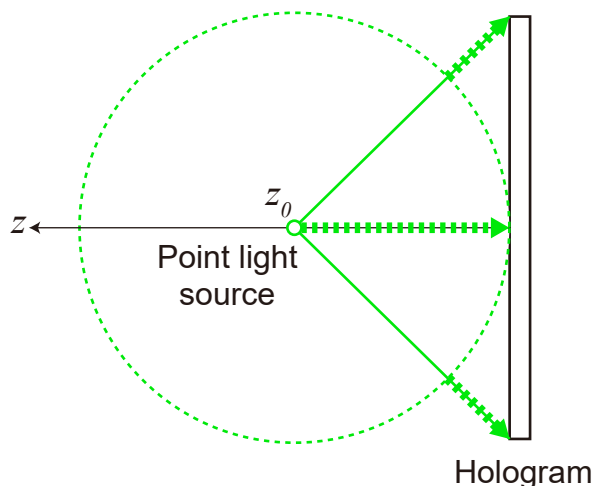


図 1.5: 点光源 (球面波) の半径 z_0 の波面。

そうすると、式 (1.4) は式 (1.14) で表される。

$$r = \sqrt{x_a^2 + y_a^2 + z_0^2} - z_0 \quad (1.14)$$

ここで $z_0 \gg x, y$ とし、テイラー展開により 1 次までの項を取ると、式 (1.15) のように求められる。

$$\begin{aligned} \sqrt{x_a^2 + y_a^2 + z_0^2} &= z_0 \left\{ 1 + \left(\frac{x_a^2 + y_a^2}{z_0^2} \right) \right\}^{\frac{1}{2}} \\ &\approx z_0 \left\{ 1 + \frac{1}{2} \left(\frac{x_a^2 + y_a^2}{z_0^2} \right) \right\} \\ &= z_0 + \left(\frac{x_a^2 + y_a^2}{2z_0} \right) \end{aligned} \quad (1.15)$$

式 (1.15) から、式 (1.14) は式 (1.16) で表される。

$$r = \sqrt{x_a^2 + y_a^2 + z_0^2} - z_0 \approx \frac{x_a^2 + y_a^2}{2z_0} \quad (1.16)$$

複数点によるホログラムを考えたとき、式 (1.16) を利用して求めるホログラムは $z_0 \gg x, y$ より $\frac{1}{r}$ を $\frac{1}{z_0}$ で近似して定数として省略し、式 (1.17) を新たに定義すると、振幅型ホログラムは式 (1.18)、位相型ホログラムは式 (1.19) で求められる。

$$\theta_{aj} = \left[\frac{1}{2\lambda |z_j|} \{ (x_a - x_j)^2 + (y_a - y_j)^2 \} \right] \quad (1.17)$$

$$I(x_\alpha, y_\alpha) = \sum_{j=0}^{M-1} \cos(2\pi\theta_{aj}) \quad (1.18)$$

$$\phi(x_\alpha, y_\alpha) = \tan^{-1} \left\{ \frac{\sum_{j=0}^{M-1} \sin(2\pi\theta_{aj})}{\sum_{j=0}^{M-1} \cos(2\pi\theta_{aj})} \right\} \quad (1.19)$$

この計算式には開平計算が含まれておらず、その分計算の高速化が可能である。

1.4.2 漸化式法を用いた計算

CGH を表示する LCD は、一般的に画素が等間隔で並んでいる。この特徴を利用すると、ある点光源から CGH 上の画素への光波を計算したあと、隣り合う画素に関して同じく光波を求めるのではなく、事前に必要な差分を求め、差分を単純に加算することで計算できる [20]。

式 (1.17) より、ホログラム面上の任意の画素 (x_a, y_a) から x 軸方向に 1 画素離れた (x_{a+1}, y_a) の位相を $\theta_{(a+1)j}$ とすると、 $\theta_{(a+1)j}$ は式 (1.20) で表される。ここで、 $x_{aj} = x_a - x_j$, $y_{aj} = y_a - y_j$ である。

$$\begin{aligned}\theta_{(a+1)j} &= \frac{1}{2\lambda z_j} \{(x_{aj} + 1)^2 + y_{aj}^2\} \\ &= \frac{1}{2\lambda z_j} (x_{aj}^2 + y_{aj}^2) + \frac{1}{2\lambda z_j} (2x_{aj} + 1) \\ &= \theta_{aj} + \frac{1}{2\lambda z_j} (2x_{aj} + 1)\end{aligned}\quad (1.20)$$

(x_a, y_a) と (x_{a+1}, y_a) の差分を式 (1.21) で表す。また、式 (1.22) を新たに定義する。ここで、 $\rho_j = 1/2\lambda z_j$ である。

$$\Delta_{aj} = \frac{1}{2\lambda z_j} \{2x_{aj} + 1\} = \rho_j \{2x_{aj} + 1\} \quad (1.21)$$

$$\Gamma_j = \frac{1}{\lambda z_j} = \frac{2}{2\lambda z_j} = 2\rho_j \quad (1.22)$$

x 軸の隣り合う画素に関して続けて計算していくと、次のようになる。

$$\begin{aligned}\theta_{(a+2)j} &= \rho_j \{(x_{aj} + 2)^2 + y_{aj}^2\} \\ &= \rho_j (x_{aj}^2 + y_{aj}^2) + \rho_j (4x_{aj} + 4) \\ &= \theta_{aj} + \rho_j (2x_{aj} + 1) + \rho_j (2x_{aj} + 1) + 2\rho_j \\ &= \theta_{(a+1)j} + \rho_j (2x_{aj} + 1) + 2\rho_j \\ &= \theta_{(a+1)j} + \Delta_{aj} + \Gamma_j \\ \theta_{(a+3)j} &= \rho_j \{(x_{aj} + 3)^2 + y_{aj}^2\} \\ &= \rho_j (x_{aj}^2 + y_{aj}^2) + \rho_j (6x_{aj} + 9) \\ &= \theta_{aj} + \rho_j (2x_{aj} + 1) + \rho_j (2x_{aj} + 1) + 2\rho_j + \rho_j (2x_{aj} + 1) + 4\rho_j \\ &= \theta_{(a+2)j} + \rho_j (2x_{aj} + 1) + 4\rho_j \\ &= \theta_{(a+2)j} + \Delta_{aj} + 2\Gamma_j \\ \theta_{(a+n)j} &= \rho \{(x_{aj} + n)^2 + y_{aj}^2\} \\ &= \rho (x_{aj}^2 + y_{aj}^2) + \rho (nx_{aj} + n^2) + \rho \\ &= \theta_{(a+n-1)j} + \rho (2x_{aj} + 1) + 2\rho (n-1) \\ &= \theta_{(a+n-1)j} + \Delta_{aj} + (n-1)\Gamma_j\end{aligned}\quad (1.23)$$

式 (1.23) のように漸化式で表現できる手法を漸化式法と呼ぶ。

以上より、漸化式法により CGH を計算するときは、始点に関して式 (1.17) を計算すると同時に、式 (1.21) と式 (1.22) を求める。CGH 上の x 方向の隣り合う画素

n に関して, θ_{nj} は式 (1.24) で計算する.

$$\theta_{nj} = \theta_{(n-1)j} + \Delta_{(n-1)j} \quad (1.24)$$

このとき, Δ_{nj} を式 (1.25) により更新する.

$$\Delta_{nj} = \Delta_{(n-1)j} + \Gamma_j \quad (1.25)$$

x 軸方向の隣り合う θ_{nj} に関して, 式 (1.24) と式 (1.25) の計算を繰り返すことにより CGH を求めることが可能である. 漸化式法では, 隣り合う画素に関して, 式 (1.17) の計算を繰り返す代わりに, 単純な加算を繰り返すことで計算可能である (図 1.6). この特徴によって, 特にハードウェア実装したとき, より少ない演算器で CGH を計算可能な回路を構成できる. 必要なリソース数を削減し, 計算コアの並列数を増加可能な利点がある.

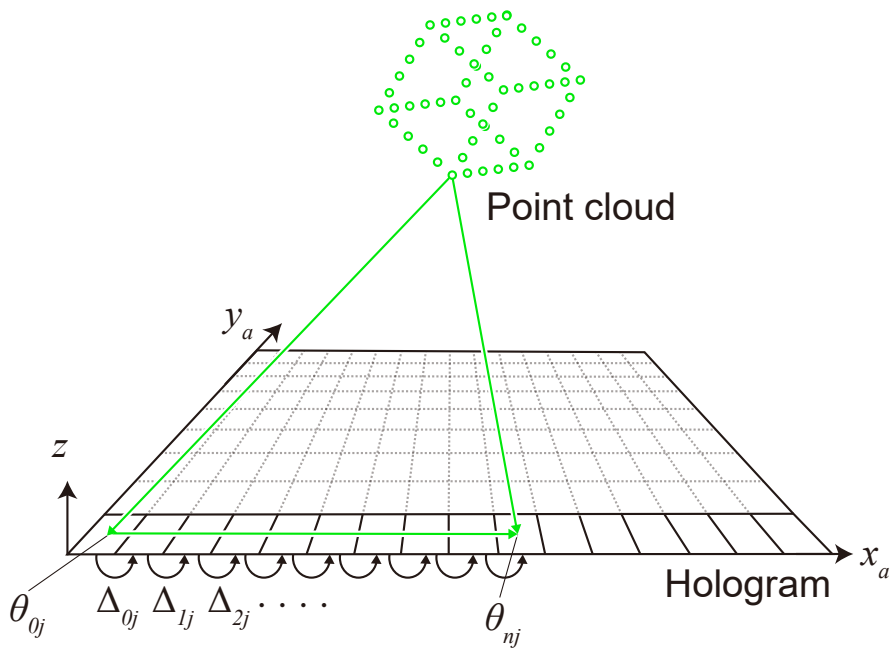


図 1.6: 漸化式法の適用イメージ.

1.5 デジタルホログラフィの原理

デジタルホログラフィは, ホログラムを Charge Coupled Device (CCD) や Complementary Metal Oxide Semiconductor (CMOS) などの撮像素子で撮影し, ホログラムを画像データとして計算機に保存する. 計算機内でホログラムから撮影物体の

再生像を数値的に計算し、1枚のホログラムデータから任意の奥行きに焦点を合わせた再生像を計算可能である。撮影素子を利用することで、静止した物体だけでなく動体も3次元的に観察できる。デジタルホログラフィによって、物体の3次元情報の記録および定量的解析が可能になった [9-13]。

1.5.1 フレネル-キルヒホッフの回折積分公式

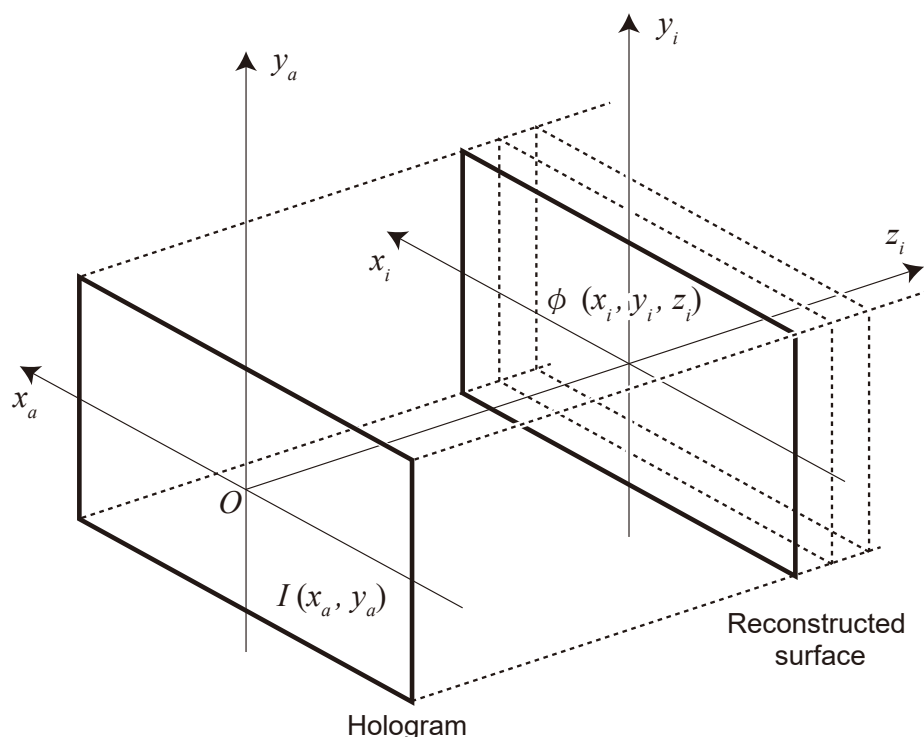


図 1.7: フレネル-キルヒホッフの回折積分公式による再生像計算。

数値計算的にホログラムから再生を行う場合、ホログラムの干渉縞を回折格子として捉え、回折計算を行う。ホログラム上のある点 $(x_a, y_a, 0)$ から再生空間上のある点 (x_i, y_i, z_i) での光波は、参照光と物体光が平行に入射されている場合、フレネル-キルヒホッフの回折積分公式によって式 (1.26), 式 (1.27) で表される [44, 61]。ここで、 $\phi(x_i, y_i, z_i)$ は再生空間の複素振幅、 $I(x_a, y_a)$ はホログラム面上の光の強度、 λ は光の波長、 k は $k = 2\pi/\lambda$ で表される光の波数、 N はホログラム面と再生面の縦と横の範囲をそれぞれ表す (図 1.7)。

$$\phi(x_i, y_i, z_i) = \frac{1}{i\lambda} \int_{-N/2}^{N/2} \int_{-N/2}^{N/2} I(x_a, y_a) \frac{\exp(ikr_{ai})}{r_{ai}} \frac{z_i}{r_{ai}} dx_a dy_a \quad (1.26)$$

$$r_{ai} = \sqrt{(x_a - x_i)^2 + (y_a - y_i)^2 + z_i^2} \quad (1.27)$$

1.6 デジタルホログラフィ計算の高速化手法

フレネル-キルヒホッフの回折積分公式を利用したデジタルホログラフィ計算は、 $N \times N$ 画素の場合、 $O(N^4)$ の計算量が必要である。フレネル近似を利用した近似計算や高速フーリエ変換を利用した計算の高速化が行われている。

1.6.1 フレネル近似を用いた計算

デジタルホログラフィでは、テイラー展開を利用した近似計算が行われている。式 (1.27) の \exp 項の中の距離 r_{aj} は、テイラー展開を利用して式 (1.28) のように近似できる。

$$\begin{aligned} r_{ai} &= z_i \sqrt{1 + \frac{(x_a - x_i)^2 + (y_a - y_i)^2}{z_i^2}} \\ &\cong z_i + \frac{(x_a - x_i)^2 + (y_a - y_i)^2}{2z_i} - \frac{\{(x_a - x_i)^2 + (y_a - y_i)^2\}^2}{8z_i^3} + \dots \end{aligned} \quad (1.28)$$

式 (1.28) の第2項までを利用した近似手法をフレネル近似と呼ぶ。また、 \exp 項の中の距離 r_{ai} は光の位相に関するため式 (1.28) により比較的正確に計算する必要があるが、その他の r_{ai} は振幅のみに影響するので $r_{ai} \approx z_i$ としても差し支えない [44]。よって、 $(x_a - x_i)^2 + (y_a - y_i)^2 \ll z_i^2$ の条件下でこれらの近似を利用し、式 (1.26) は式 (1.29) で表される。

$$\begin{aligned} \phi(x_i, y_i, z_i) &\approx \frac{1}{i\lambda z_i} \int_{-N/2}^{N/2} \int_{-N/2}^{N/2} I(x_a, y_a) \exp \left[ikz_i \left\{ 1 + \frac{1}{2} \left(\frac{x_a - x_i}{z_i} \right)^2 \right. \right. \\ &\quad \left. \left. + \frac{1}{2} \left(\frac{y_a - y_i}{z_i} \right)^2 \right\} \right] dx_a dy_a \\ &= \frac{\exp(ikz_i)}{i\lambda z_i} \int_{-N/2}^{N/2} \int_{-N/2}^{N/2} I(x_a, y_a) \exp \left[\frac{ik}{2z_i} \{(x_a - x_i)^2 \right. \\ &\quad \left. + (y_a - y_i)^2\} \right] dx_a dy_a \end{aligned} \quad (1.29)$$

1.6.2 畳み込み回折積分法を用いた計算

式 (1.29) の計算量は、ホログラム全体の画素数 × 再生面全体の画素数となる．実際に計算機で計算する場合，計算時間の観点から畳み込み定理を用いる．1つの再生面において z_i は定数と見なせ，ある再生距離 z_i におけるホログラムによる再生像の複素振幅 $\phi(x_i, y_i)$ は式 (1.30)，式 (1.31) で表される．

$$\phi(x_i, y_i) = \int_{-N/2}^{N/2} \int_{-N/2}^{N/2} I(x_a, y_a) g(x_i - x_a, y_i - y_a) dx_a dy_a \quad (1.30)$$

$$g(x_i - x_a, y_i - y_a) = \frac{\exp(ikz_i)}{i\lambda z_i} \exp\left[\frac{ik}{2z_i} \{(x_i - x_a)^2 + (y_i - y_a)^2\}\right] \quad (1.31)$$

式 (1.30) は，x 軸および y 軸についての二次元畳み込み積分であり，二次元フーリエ変換によって変換することができ，式 (1.32) で表される [44, 61]．

$$\Phi(n, m) = \hat{I}(n, m)G(n, m) \quad (1.32)$$

ここで， $\Phi(n, m)$ は $\phi(x_i, y_i)$ の二次元フーリエ変換である．同様に， $\hat{I}(n, m)$ は $I(x_\alpha, y_\alpha)$ ， $G(n, m)$ は $g(x_i - x_\alpha, y_i - y_\alpha)$ のフーリエ変換を表す． $I(x_\alpha, y_\alpha)$ と $g(x_i - x_\alpha, y_i - y_\alpha)$ の畳み込み積分は，それぞれフーリエ変換を行うと，フーリエ変換後の領域では， $\hat{I}(n, m)$ と $G(n, m)$ の積で表すことができる．積の計算結果を逆二次元フーリエ変換し，元の領域に戻すことで再生面の複素振幅を得ることができる．

ここで，式 (1.30) と式 (1.32) の計算量について考える．式 (1.30) をそのまま計算した場合， $O(N^4)$ の計算量が必要であった．一方で式 (1.32) では，高速フーリエ変換 (Fast Fourier Transform: FFT) を用いることにより，計算量の増加を抑えることができる．二次元FFTに必要な計算量は $O(N^2 \log_2 N)$ で表され，大幅に計算量を減らすことが可能である．FFTを使った場合と使わなかった場合の計算量の比は，式 (1.33) で表される [44]．

$$\frac{N^2 \log_2 N}{N^4} = \frac{\log_2 N}{N^2} \quad (1.33)$$

FFTを使った計算量削減では，画素数 N が多いほど効果は大きい．本研究では， $N = 128 (= 2^7)$ と比較的小さな画素数で利用するが，それでも計算量の比は $7/128^2 \approx 1/10^4$ となり，1万分の1程度まで計算量を削減可能である．

式 (1.31) の $\frac{\exp(ikz_i)}{i\lambda z_i}$ と $G(n, m)$ を乗じたものをフーリエ変換 \mathcal{F} は解析的に求めることができ，式 (1.34) で表される [44]．ここで，ホログラム面および再生面の画素間隔を ΔP ，ホログラムの画素数を N ， $\Delta f = 1/(N\Delta P)$ を周波数領域のサンプリン

グ間隔と置いた.

$$\begin{aligned}
 G(n\Delta f, m\Delta f) &= \mathcal{F} \left[\frac{\exp(ikz_i)}{i\lambda z_i} G(n, m) \right] \\
 &= \exp(ikz_i) \exp \left\{ 2\pi i \left(\frac{-\lambda z_i}{2N^2(\Delta P)^2} \right) (n^2 + m^2) \right\} \quad (1.34)
 \end{aligned}$$

本研究では, 再生距離 (奥行き) の異なる再生像を重ねて計測対象の 3 次元情報を再生する. したがって, 1 つの面を再生しているときには $\exp(ikz_i)$ の項は定数とみなすことができる. また, $\phi(x_i, y_i)$ の最小値と最大値をもとにして 256 階調化しするため, $\exp(ikz_i)$ は省略することができる. したがって, 実際に再生像を計算する場合は式 (1.35) を利用する.

$$G(n\Delta f, m\Delta f) = \exp \left\{ 2\pi i \left(\frac{-\lambda z_i}{2N^2(\Delta P)^2} \right) (n^2 + m^2) \right\} \quad (1.35)$$

第2章 VC707を用いたデジタルホログラフィ専用計算機

デジタルホログラフィで必要とされる再生像の計算時間削減のため、書き換え可能な Large Scale Integration (LSI) である Field Programmable Gate Array (FPGA) を用いて、計算処理を高速に行う専用計算機の開発が行われている [16,17]. デジタルホログラフィ計算において、特に高速フーリエ変換時の計算負荷が大きい. 高速フーリエ変換では、バタフライ演算を繰り返す単純な計算なため、ハードウェア化に適している. 先行研究では、FFT 計算をハードウェアで計算することにより、効率的な計算が可能であることが示された [16,17]. 一方で、外部メモリを利用した構成のため、メモリ転送にかかるオーバーヘッドにより、さらなる高速化は困難であった.

本研究では、ハイスピードカメラを利用したデジタルホログラフィ向け専用計算機の開発を行った. カメラは高速になればなるほど撮像素子の画素数が小さくなる. 128×128 画素のホログラムを対象として、内部メモリだけで計算可能なシステムの開発を行った. その結果、1 フレームのホログラムから再生距離 (奥行き) の異なる 8 フレームの再生像計算を行うユニットを 4 つ搭載し、計 32 枚の再生像を 0.69 ms で計算可能なシステムの開発に成功した. 開発した専用計算機は 4 ユニットで、6 コアの CPU や 2,560 CUDA コアの GPU と比べ 23 倍の高速化を実現した.

2.1 デジタルホログラフィ専用計算機 FFT-HORN の開発

先行研究では、デジタルホログラフィ向け専用計算機として Fast Fourier Transform Holographic ReconstructioN (FFT-HORN) の開発が行われた. FFT-HORN は、流体の可視化を目的とした Digital Holographic Particle Tracking Velocimetry (DHPTV) 向けに $1,024 \times 1,024$ 画素を超えるホログラムが計算可能なものが開発された [16,17]. デジタルホログラフィ計算では計算に必要な Fast Fourier Transform (FFT) 計算にかかる時間が支配的である. FFT 計算をハードウェアで計算することにより、効率的な計算が可能であることが示された [16,17].

一方で、FFT-HORNでは $1,024 \times 1,024$ 画素のホログラム計算のため、外部メモリを利用した構成のため、メモリ転送のオーバーヘッドが存在した。専用計算機では、計算に必要なデータのすべてを専用計算機内部でキャッシュできた方がデータ転送にかかる時間を削減でき、より高速化が期待できる。本研究では、専用計算機の対象を比較的小さな画素数となるハイスピードカメラ向けに絞り、専用計算機の開発を行った。カメラは高速になればなるほど撮像素子の画素数が小さくなる。 128×128 画素のホログラムを対象として、すべてのデータを内部メモリのみで実装した計算機の開発を行った。

2.2 デジタルホログラフィ専用計算機の構成

高速イメージング向けデジタルホログラフィ専用計算機開発の初期段階として、Xilinx社が提供するVirtex-7 FPGA VC707評価キット(以下、VC707と呼ぶ)を用いた。VC707は、Virtex-7 XC7VX485T-2FFG1761CのFPGAを搭載した評価ボードである。VC707の外観と搭載されているFPGAの仕様をそれぞれ表2.1、図2.1に示す。

表2.1において、Look-up tableは参照テーブルにより、所望の論理関数を構成する論理回路である。Flip flopはビット情報を一時的に保持する論理回路である。回路は基本、Look-up tableとFlip flopを組み合わせて構成する。Block RAMはFPGA内部のメモリである。計算に必要なデータや計算結果を一時保存するなど、キャッシュとして利用する。

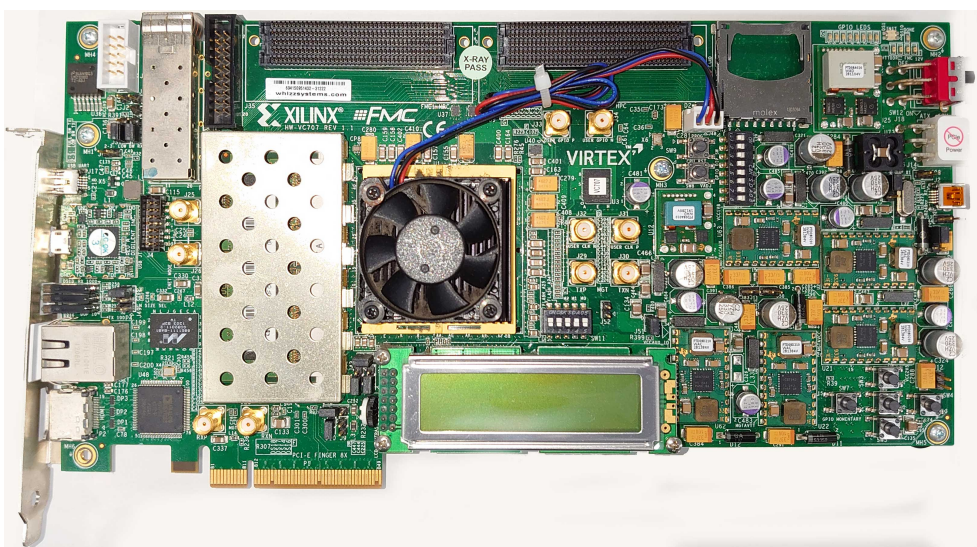


図 2.1: Virtex-7 FPGA VC707 評価キットの外観. 黒色のファンの下にFPGAチップが搭載されている。

表 2.1: VC707 の論理回路リソース.

Resource	Available
Look-up table	303,600
Flip flop	607,200
Block RAM	37,080 Kb

デジタルホログラフィ向け専用計算機システムの概略図を図 2.2 に示す. VC707 は評価ボードであり, 様々なインターフェイスを備えている. 本章のシステムは PCI Express を利用し, PC と接続して利用する構成とした. PC (CPU) 側から計算アクセラレータとして, VC707 内部に構築した専用回路を利用することで, 計算を高速化した.

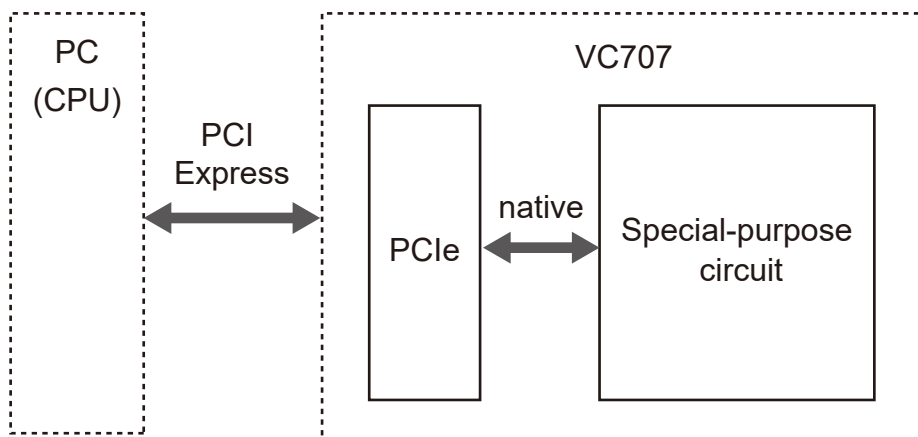


図 2.2: デジタルホログラフィ専用計算機システム全体のブロック図.

VC707 では, Xilinx 社の 7 シリーズ向けに提供されている IP コア (Intellectual Property Core) である 7 Series Integrated Block for PCI Express (PCIe) [62] を利用した通信が可能である. PCIe は, Programmable I/O (PIO) を用いた低速な通信ではあるが, Block RAM にデータを読み書きするリファレンスデザインが提供されており, 比較的容易に計算回路開発へ取りかかることが可能である. PCIe では, そのまま Block RAM へ接続可能なアドレス線, データ線, 読み書きイネーブル線からなるネイティブプロトコル (Block RAM へ読み書きする基本プロトコル) による通信が可能である. 計算回路とは別に通信用のユニットを作る必要は無く, そのまま計算回路内部の Block RAM へ接続可能である. 本章では, 研究の初期段階として, PCIe を利用した通信回路を利用した.

2.3 デジタルホログラフィ専用計算回路の実装

畳み込み回折積分法を利用した計算回路をFPGA上に実装した。ハイスピードカメラを用いたデジタルホログラフィ向け専用計算機として、 128×128 画素のホログラムの計算を行う専用計算機を開発した。実装した回路のブロック図を図2.3に示した。

式(1.30)、式(1.31)に示した畳み込み回折積分法を利用し、回折計算を行うユニットを4つ搭載した。それぞれのユニットは、1フレームのホログラムから再生距離の異なる8フレームの再生像計算を行う。専用計算機全体で、4フレームのホログラムから、32フレームの再生像計算を行う専用計算機を開発した。

各ユニットは、ホログラムや計算パラメータを保存するBlock RAMと式(1.32)、式(1.35)に示した2次元畳み込み回折計算を行うユニットから構成した。Block RAMはTrue Dual Port RAMであり、読み書きするポートがそれぞれ2つずつある。一組のポートをPCIeと繋ぎ、もう一組は回折計算を行う計算回路へつなぐことで、CPUと計算回路の通信を可能とした。

2.3.1 2次元畳み込み計算回路の実装

図2.4に畳み込み回折積分法による専用計算パイプラインのブロック図を示す。図2.4は、式(1.30)、式(1.31)に示した計算を行う。図中の斜線上の数字は、データのbit幅を表す。図2.4の回路は、データ保存用のRAM、FFT、Inverse Fast Fourier Transform (IFFT)を行うユニット、式(1.35)を計算するGユニットから構成した。この回路では、ホログラムのフーリエ変換と逆変換を行うのに、Xilinx社が提供するIPコア(Fast Fourier Transform v7.1 [63])を使用した。このコアは、入力により1次元のFFTと逆FFTを選択することが出来る。本研究では、2次元のFFTと逆FFTを行う必要があるため、2次元FFTを行うときにはこのコアを用いて、まず水平方向の1次元FFTを行った後、垂直方向の1次元FFTを行った。逆FFTについても同様である。マルチプレクサ(MULTipleXer: MUX)を利用し、1つのFFT IPコアを使い回すことにより、リソースの削減を行っている。

FFTおよび逆FFTの計算では、計算結果が入力値よりも大きくなる可能性がある。このため、コアの内部でオーバーフローが発生する可能性がある。これを防ぐために、場合によってはFFTの結果に $1/2$ 、 $1/4$ 、 $1/8$ など1以下の係数を掛ける必要がある。今回用いたFFTのコアでは、どのタイミングでどの係数を掛けるかを指定することができる。この指定はスケーリング・スケジュール(scale_sch)と呼ばれている。最適なスケーリング・スケジュールはホログラムに依存するため、今回はホストPCからBlock RAMへ値を書き込みことでスケーリング・スケジュールを設定できるようにした。

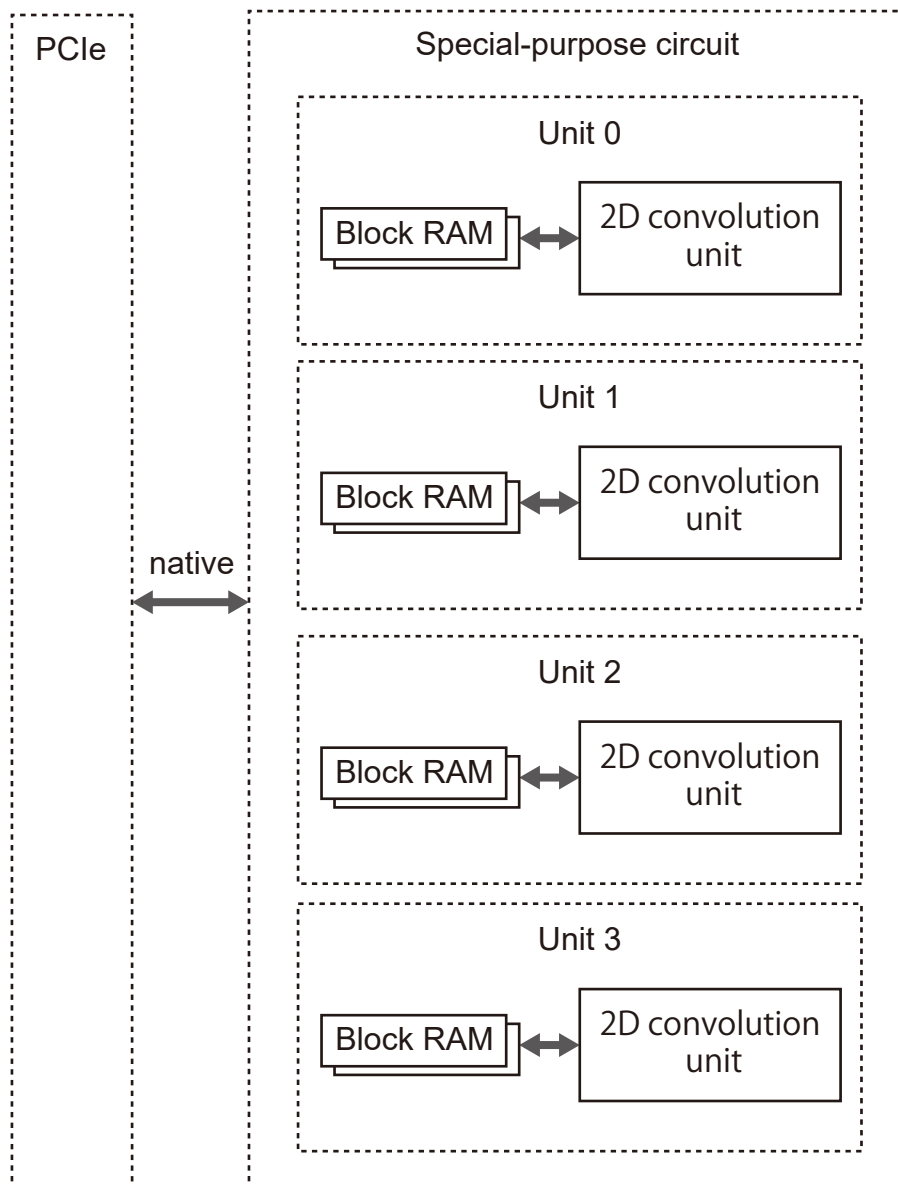


図 2.3: デジタルホログラフィ専用計算回路のブロック図 (VC707 内部).

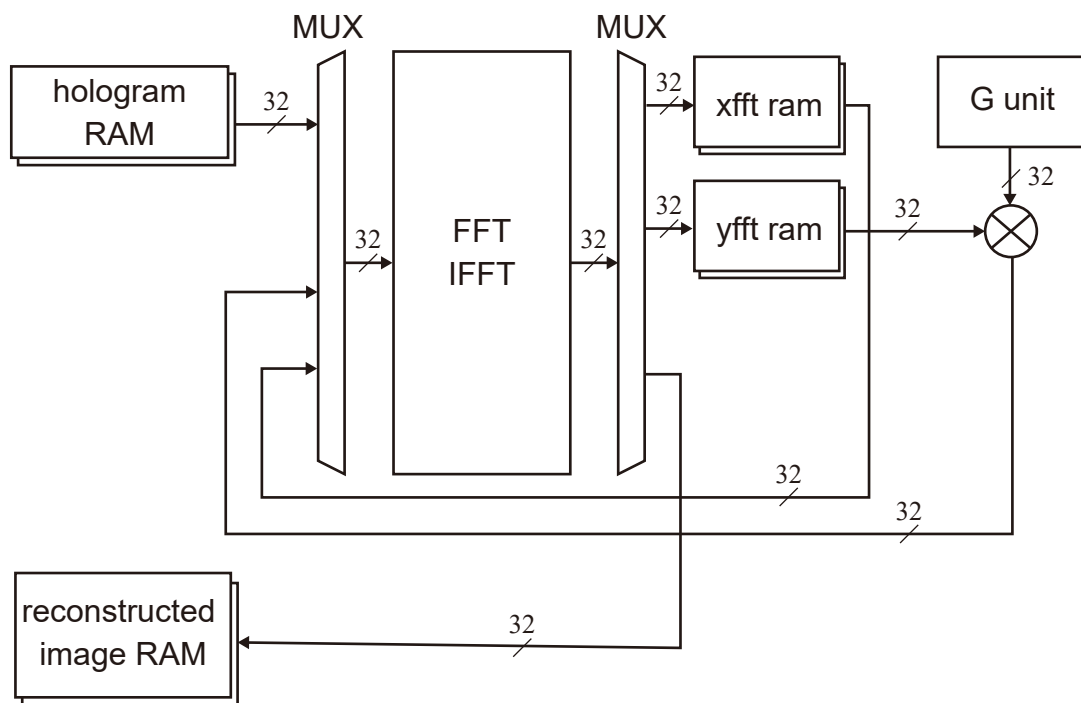


図 2.4: 2次元畳み込み計算ユニットのブロック図.

ホログラム面と再生面の距離 z_i は、式 (1.35) の $G(n, m)$ にのみ影響する。したがって、それぞれの再生面に対応した距離 z_i をパラメータとして与えた $G(n, m)$ を1枚のホログラムのフーリエ変換 $\hat{I}(n, m)$ にそれぞれ乗算を行うことで複数の奥行方向の再生面強度を求めることが可能となる。

専用パイプラインは4つのステージを用いたステートマシンとなっている。すなわち、以下に示す4ステージにより、1枚のホログラムから8枚の再生距離の異なる再生像を計算する。

- ステージ1: ホログラムデータを hologram RAM から読み込み、1次元FFT (水平方向) を行い、結果を図 2.4 中の xfft ram へ保存する。これを垂直方向128ライン分繰り返す。
- ステージ2: ステージ1におけるFFTの計算結果を xfft ram から垂直方向を読み込み、1次元FFTを行い、図 2.4 中の yfft ram へ保存する。これを水平方向128ライン分繰り返す。
- ステージ3: ステージ2におけるFFTの計算結果を yfft ram から読み込み、 G の値と乗算する。その後、1次元逆FFTを行い、結果を図 2.4 中の xfft ram へ保存する。これを垂直方向128ライン分繰り返す。
- ステージ4: ステージ3における逆FFTの計算結果を xfft ram から垂直方向を読み込み、1次元逆FFTを行い、図 2.4 中の reconstructed image RAM へ保存する。これを水平方向128ライン分繰り返す。

ステージ3とステージ4を再生距離 z_i に間隔 dz を加算しつつ繰り返す、8再生面分計算が完了するまで続ける。

FFTを使用した畳み込み計算実装の注意点として、循環畳み込みと呼ばれる、回り込みが計算結果に重畳されることに注意が必要である。これを避けるために、一般的にはゼロパディング (ゼロ埋め) を利用した回折計算が行われているが、本研究では利用していない。ゼロパディングを行う場合、平均値処理を行う必要がある。しかし、平均値処理にはある程度計算時間が必要である。また、メモリを消費し、FPGAリソース的に不利である。平均値処理を行わず、ゼロパディングを行うことも可能だが、強いリングングと呼ばれるノイズが発生する。ゼロパディングを行わない場合、本稿の計算式ではサンプリング条件を満たす z_i は式 (2.1) で表される。

$$z_i = \frac{N \times \Delta P^2}{\lambda} \quad (2.1)$$

サンプリング条件を見たさない場合、多少のエリアシングが発生するが、リングングは発生しない。強度画像観察するのみであれば十分と考えられる。本研究では、高速計算する上で有利な計算コストの削減とFPGAメモリ削減のため、ゼロパディングを利用しない構成とした。

2.3.2 G 計算ユニットの実装

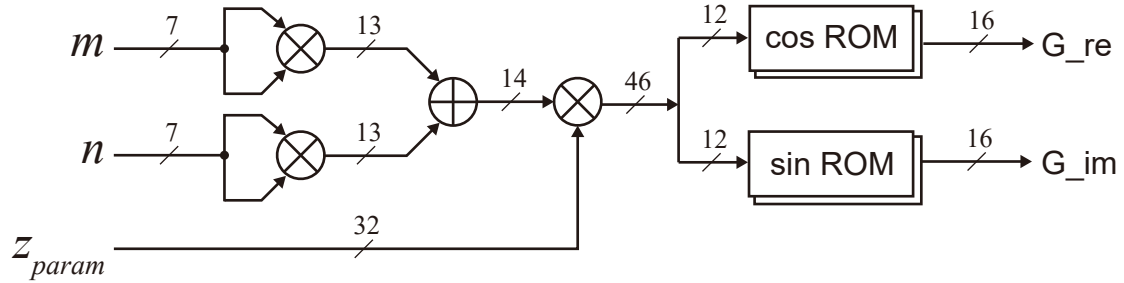


図 2.5: G unit のブロック図.

図 2.5 に図 2.4 中, G の計算をする G ユニットのブロック図を示す. 図 2.5 は, 式 (1.35) を計算するユニットである. ここで, 新たに奥行関連係数として式 (2.2) を定義する.

$$z_{param} = -\frac{\pi \lambda z_i}{2N^2 (\Delta P)^2} \quad (2.2)$$

すると, 式 (1.35) は式 (2.2) を用いて, 式 (2.3) で表される.

$$G(n, m) = \exp \{2\pi i z_{param} (n^2 + m^2)\} \quad (2.3)$$

オイラーの公式より, $\exp(i\theta) = \cos \theta + i \sin \theta$ が成り立つ. この θ を計算した後, $\cos \theta$ の値が格納されたテーブル cos ROM と $\sin \theta$ の値が格納されたテーブル sin ROM を用いて, $G(n, m)$ の実数部と虚数部の計算をした. 式 (2.3) の指数は $2\pi i$ で規格化されているため, $z_{param} (n^2 + m^2)$ の指数部は必要なく, その小数点以下のみ計算した. 回路に入力されている z_{param} は, $-\lambda z_i / 2N^2 (\Delta P)^2$ の小数点以下 32 ビットとした. また, この項の計算は事前に計算可能であるため, 任意の再生距離 z_i の初期値 z_0 と再生距離間隔 dz を用いてホスト PC によって計算し, FPGA へ転送することで FPGA リソースの削減, 動作周波数の向上を図った.

2.4 リソース使用率

VC707 で開発した専用計算機の回路規模について表 2.2 示す. 動作周波数は, 250 MHz で動作する.

表 2.2: デジタルホログラフィ専用計算機のリソース利用率.

Resource	Number of usage	Usage[%]
Look-up table	13,283	4
Flip flop	16,460	2
Block RAM	24,844 Kb	67

表 2.2 より, Flip flop と Look-up table については回路リソースに余裕があることが分かる. 本計算システムは計算コアを 4 つ備えており, 各計算コアにそれぞれ Block RAM を配置しているため, Block RAM を増設する場合, 4 倍のリソース量を必要とする. したがって, Block RAM 利用率は 67% と余裕があるようにも見えるが, これ以上の大幅な Block RAM の増設は困難であり, ユニットの並列数は 4 が上限となった.

2.5 計算性能比較

計算時間比較のため, CPU と Graphics Processing Unit (GPU) でも 128×128 画素の 4 枚のホログラムから 32 枚の再生像を計算するプログラムを作成した. 計算時間の比較を表 2.3 に示した [43]. CPU は Intel Core i5-8400 CPU 2.80GHz (6 コア), Ubuntu 18.04.4 (Linux Kernel 4.15.0-96) が動作するシステムで, Intel C compiler 19.1.0.166 を用いた. FFT の計算には FFTW3.3.8 を利用した. 表 2.3 に, 1 コアでの計算時間と 6 コアでの計算時間を示した. GPU は NVIDIA GeForce GTX 1080 (2,560 CUDA コア) を利用した. NVIDIA CUDA compiler 9.1 を利用しプログラムを作成した. FFT の計算には NVIDIA CUDA compiler 9.1 に含まれる cuFFT ライブラリ [64] を利用した. 表 2.3 に計算時間を示した. 開発した専用計算機はシングルコアの CPU と比べ 36 倍の高速化を達成した. 6 コアの CPU や 2,560 CUDA コアの GPU と比べても 23 倍の高速化を実現した.

表 2.3: デジタルホログラフィ再生計算時間の比較.

Hardware	Calculation time [ms]	Acceleration ratio
FPGA: VC707	0.69	36.36
GPU: NVIDIA GeForce GTX 1080 (2,560 CUDA Cores)	15.90	1.58
CPU: Intel Core i5-8400 (6 Cores)	16.50	1.52
CPU: Intel Core i5-8400 (1 Cores)	25.09	1.00

表 2.3 において, 1 コアで計算した場合と, 6 コアで計算した場合で 1.52 倍しか高速化されていないことがわかる. また, CPU を使用し 6 コアで計算した場合と, 2,560 CUDA コアの GPU で計算した場合で, あまり計算時間に差が無いことがわかる. 一般的に, マルチコアシステムは並列実行に事前処理や制御などのオーバーヘッドが存在する. 並列度を上げてても並列度に合わせた高速化が得られないことが知られている. 今回は特に高速度イメージング向けに 128×128 画素のホログラムを扱った. 画素数が小さいため, 1 枚の処理は即座に終了する. 表 2.3 でも数十 ms で計算が終了している. この場合, 並列実行にかかるオーバーヘッドがかえって顕在化し, 高速化は頭打ちとなる. この問題は高速度 3 次元イメージングでは並列計算による高速化が得られにくいことを示す. 高速度 3 次元イメージングでは, 小さい画素数でフレーム数が数万から数億にわたる処理が必要なため, 一般的な計算機では高速化が難しくなる.

一方で, 専用計算機ではそのようなオーバーヘッドは存在せず, 表 2.3 では 4 ユニットで 23 倍の高速化を達成している. 本研究の結果は, 高速度 3 次元イメージングといった小さい画素数でフレーム数が数万から数億にわたる処理での専用計算機の有用性を示唆している.

2.6 再生像評価

図 2.6 に示す光学系を構築し、ホログラムの撮影を行った。光源として、532 nm のレーザーを利用した。ホログラムはグラニュー糖が落下する様子を 1/300,000 秒の露光時間で撮影した。図 2.7 は、CPU により倍精度で計算した再生像と開発した専用計算機で計算した再生像の比較である [43]。 $z_i = 0.21$ m と $z_i = 0.26$ m それぞれの再生距離 (奥行き) での再生像を表している。 $z_i = 0.21$ m では画像下部のグラニュー糖の粒子に合焦している。 $z_i = 0.26$ m では上部の粒子に合焦している。再生条件を表 2.4 に示す。

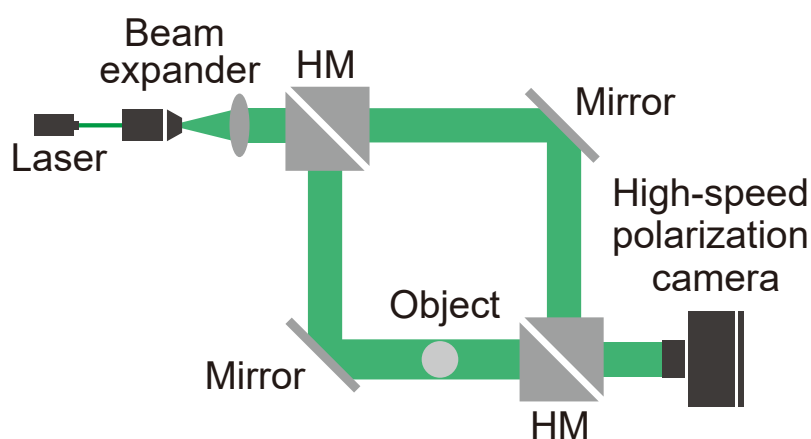


図 2.6: デジタルホログラフィ用光学系の様子。HM は Half Mirror を表す。

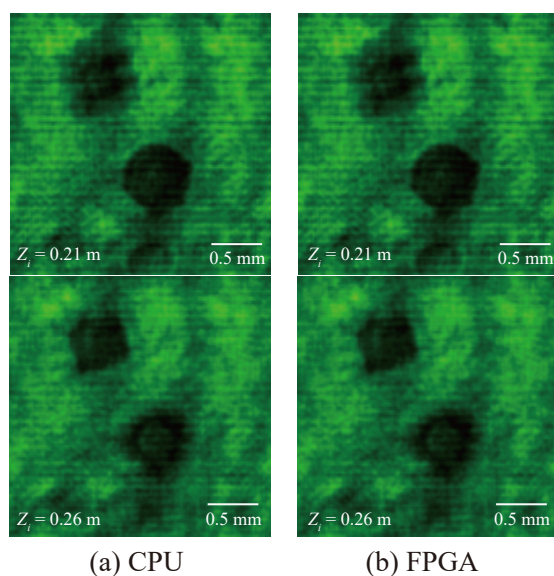


図 2.7: CPU と FPGA (VC707) による再生像計算の比較。

表 2.4: ホログラムの再生計算条件.

Parameter name	Value
Wavelength	532 nm
Pixel pitch	20 μ m
Number of pixels	128 \times 128

専用計算機が1度に計算可能な32枚分の再生像とのPSNR (Peak Signal-to-Noise Ratio) とSSIM (Structural SIMilarity) [65] の平均はそれぞれ43.8, 0.993であった。図2.7に示す通り、専用計算機ではCPUで計算した結果と同等の画質で計算できていることが分かる。

2.7 小括

本研究では、デジタルホログラフィを用いた高速度3次元イメージング向け専用計算機の開発を行った。その結果、128 \times 128画素の4枚のホログラムからそれぞれ8枚ずつ、計32枚の再生像を0.69 msで計算可能なシステムの開発に成功した。開発した専用計算機は4ユニットで、6コアのCPUや2,560 CUDA コアのGPUと比べ23倍の高速化を実現した。

一般的に、マルチコアシステムは並列実行に事前処理や制御などのオーバーヘッドが存在する。並列度を上げてても並列度に合わせた高速化が得られないことが知られている。今回扱った画素数が小さいため、並列実行にかかるオーバーヘッドがかえって顕在化し、高速化は頭打ちとなった。しかし、開発した専用計算機ではそのようなオーバーヘッドは存在せず、CPUやGPUと比べ高速化に成功した。本研究の結果は、高速度3次元イメージングといった小さい画素数でフレーム数が数万から数億にわたる処理での専用計算機の有用性を示唆するものである。

今後は計算ユニットの高密度、高集積化を行っていきたい。今回は、計算データを一時保存するBlock RAMの容量と動作周波数がボトルネックとなり4ユニットの搭載が限界であった。しかし、最新のFPGAではUltraRAMと呼ばれる大規模で高速なオンチップストレージを搭載しており、Block RAMの容量と合わせ最大で500 Mb近い容量となる。最新のFPGAボード (Xilinx ALVEO U250) を利用した場合、Look Up Table換算で100倍のリソースを搭載しているため、現状よりも100倍近い高速化が期待できる。

また、今回は専用計算機の計算時間にフォーカスしていたため、専用計算機の通信は低速なI/Oポートを利用した。通信時間は合わせて29.15 msであった。しかし、DMA転送などを利用することで通信時間は十分に小さくすることができる。デー

タの受信は計算と並行に実行することが可能なため無視でき、今後の改良で通信時間も含め高速な専用計算機の開発が可能である。

第3章 HORN-8を用いた大規模な電子ホログラフィの再生

計算機合成ホログラムを高速に生成する専用計算機にHolographic Reconstruction (HORN) [38,39]がある。1993年に発表されたHORN-1は、汎用ICを使い手配線によって計算回路の設計・実装が行われ、CGH計算における専用計算機の有効性が示された[38,39]。その後、HORN-3 [40]より、FPGAを用いた計算回路の実装が行われている。最新のHORN-8システムでは、独自に開発を行ったボードにFPGAチップを8個搭載し、8台のボードを利用した64個のFPGAチップによる並列計算を可能とする[35,36]。クラスタ全体でCGHを計算する速度は、PCの1,000倍の高速化を達成した。65,000点で構成される点光源モデルから200万画素のホログラムを60 frames per second (fps)の速さで更新可能である[35]。一方で、FPGA内部のメモリ容量の制限から、扱える点群数には制限があった。65,536 (2^{16})点が、HORN-8ハードウェアの内部メモリの制限値である。この制限は、今後FPGAを用いた専用計算機を発展させていく上で、性能限界があることを示している。

本章では、HORN-8システムへ改良を加え、CGHの表示形式に時空間分割法[66]を適用した。本来は1フレームを表す点群データを分割し、高速に切り替わる数フレームを利用し、1フレーム分の映像を提示する手法に時空間分割法がある。人間の視覚特性(残像効果)により、高速に切り替わる映像は合成され、ひとまとまりの映像として知覚される。本研究では40万点の点群を6分割し、HORN-8システムによる電子ホログラフィ再生を検証した。その結果、65,536 (2^{16})点を超える40万点クラスの点群に対してもHORN-8システムは高速に計算可能であることを示した。電子ホログラフィ向け専用計算機としては、内蔵メモリが小さい場合であっても、並列度を向上し、1フレームの計算にかかる計算時間をより小さくすることで、より高精細な3次元映像システム(電子ホログラフィ)を実現可能であることを示した。

3.1 電子ホログラフィ専用計算機HORNの開発

CGH計算高速化のため、HORNと名付けられた電子ホログラフィ専用計算機の開発が行われている[35,36,38-40,67-69]。試作機1号であるHORN-1は1992年から開発されていた[38,39]。HORN-1からHORN-2については、汎用ICを使い手配線

によって計算回路の設計・実装が行われ、CGH 計算における専用計算機の有効性が確認された。HORN-3以降は、FPGA を用いた計算回路の実装が行われている。漸化式法 [20] による CGH 計算アルゴリズムは HORN のために開発されたアルゴリズムである。HORN-4 [70] から実装され、それまで 1 つの Programmable Logic Device (PLD) に 1 つだった CGH 計算回路の並列数を飛躍的に増化させ、1 つの FPGA で 21 並列の計算を可能にした。HORN-5 は、4 個の FPGA を搭載したボードを使用し、チップあたりの並列数は 352 並列で、ボード全体では 1,408 並列を達成した [68]。HORN-6 では、HORN-5 ボードを複数台利用したクラスタシステムの開発が行われた [69]。HORN-7 では、それまで計算していた振幅型ホログラムよりも良質な再生像が得られる位相型ホログラムの計算に対応した試作機の開発が行われた。

3.2 HORN-8 システムの開発

最新の電子ホログラフィ専用計算機として、HORN-8 システムの開発が行われた [35,36]。大規模 Field-Programmable Gate Array (FPGA) を 8 個搭載した HORN-8 ボードを独自に開発した。さらに、HORN-8 ボードを 8 台用いたクラスタシステムが構築された (図 3.1)。漸化式法 [20] を用いた振幅型 CGH 計算回路をチップあたり 640 並列、ボード全体で 4,480 並列、クラスタ全体で 35,840 並列実装し、65,000 点の物体から 200 万画素のホログラムを 60 fps で生成する性能を実現した。また、HORN-8 システムを利用し、位相型 CGH を計算するシステムの開発が行われた。位相型では、チップあたり 320 並列、ボード全体で 2,240 並列、クラスタ全体で 17,920 並列実装し、65,000 点の物体から 200 万画素のホログラムを 30 fps で生成する性能を実現した。HORN-8 では、高速な計算のため、点光源データを一度、FPGA 内部に保存する必要がある。一般に FPGA 内部のメモリ量は多くなく、振幅型 HORN-8 が一度に扱える点光源数の上限は 65,536 (2^{16}) 点である。

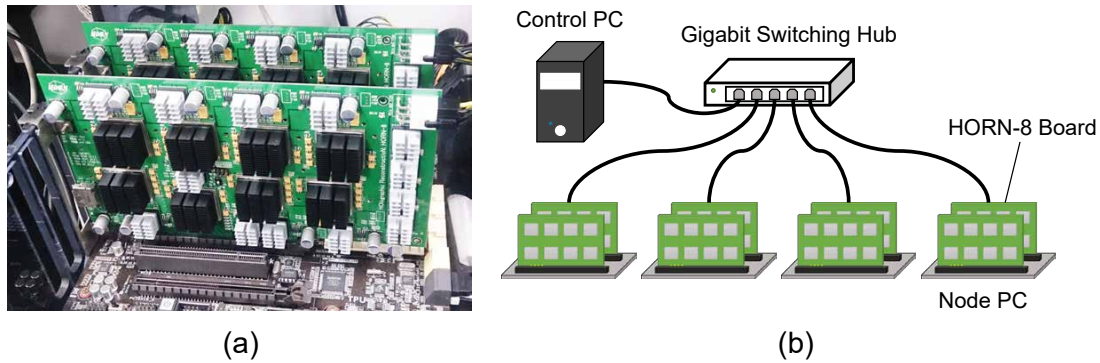


図 3.1: HORN-8 システムの外観: (a) 1 台の PC に 2 台の HORN-8 ボードを搭載している様子 (b) HORN-8 クラスタ (4 台の PC に 8 台の HORN-8 ボードを搭載している様子).

3.3 時空間分割法

電子ホログラフィには、計算負荷の問題点の他に、CGH の表示を行う空間光変調器 (Spatial Light Modulator: SLM) の性能が十分ではないという問題がある。例えば、 $1,920 \times 1,080$ 画素の空間光変調器の画素数は、200 万画素程度であり、数十万点規模の点群情報を記録した CGH の再生を高画質で行うことは困難である。空間変調器の画素数が大きくなれば高画質に再生可能な点群数は増えるが、やはり上限は存在する。

解像度が制限された SLM から高画質の電子ホログラフィ再生を行う研究が進められている。その一つに点光源を分割して再生し、目の残像効果を利用した時空間分割法がある [66]。時空間分割法では、本来は 1 フレームを表す点群データを分割し、高速に切り替わる数フレームを利用し、1 フレーム分の映像を提示する手法である。人間の視覚特性 (残像効果) により、高速に切り替わる映像は合成され、ひとまとまりの映像として知覚される。

時空間分割法を用いることで、空間変調器で本来再生可能な点群数を越えた、高画質な再生像を再生可能である。また、画質だけではなく、点群を分割することで、1 フレーム分の点群数は少なくて済む。本研究では、HORN-8 クラスタシステムに時空間分割法を適用し、 $65,536 (2^{16})$ 点の制限を超える点群 40 万点の電子ホログラフィ再生を検証した。図 3.2 に表示手法を図示した。1 秒間に 60 枚 (60 fps) の CGH の表示を行う場合、図 3.2 に示したように、点光源データを 6 分割し、それぞれから CGH を計算し、6 フレームを表示する。すなわち、60 fps で表示する場合、実際に物体が動くフレームレートは 10 fps となる [71]。

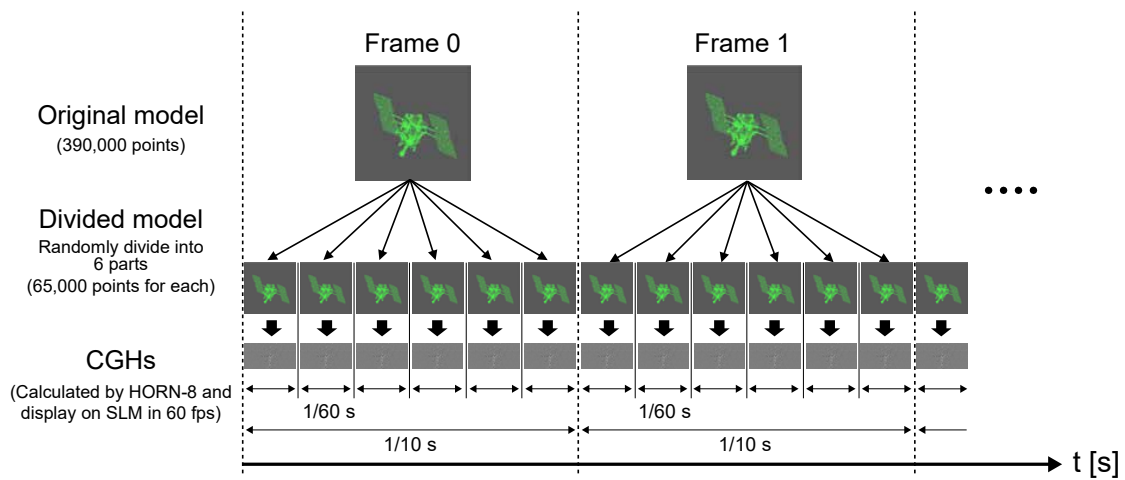


図 3.2: HORN-8 システムを用いた時空間分割法適用イメージ.

3.4 HORN-8 の性能

振幅型 HORN-8 システムの性能を示す [35]. 表 3.1 は, CPU, GPU, HORN-8 ボード単体, HORN-8 クラスタシステム (8 ボード) の比較である. 点群数は 65,000 点であり, ホログラムサイズは $1,920 \times 1,080$ 画素である.

図 3.3 は, HORN-8 クラスタシステムを用いたときの物体点数と計算時間の関係である. 物体点数が 65,000 点を超えたところでは時分割方式を用いている. HORN-8 クラスタシステムでは, 物体点数が内部メモリよりも大きくなっても, ほぼ線形に性能が向上していることがわかる [35].

表 3.1: CPU, GPU, HORN-8 ボード単体, HORN-8 クラスタシステムにおける CGH 計算時間の比較.

System	Time per hologram [ms]	Speed up ratio	Frame rate [fps]
CPU:			
Intel Core i7-6700K, 4 cores, 4 GHz	19,413	1	0.052
GPU:			
NVIDIA GTX 1080Ti, 3,584 cores, 1.48 GHz	208	90	4.8
HORN-8 board ($\times 1$):			
4,480 parallel (cores), 0.25 GHz	121	160	8.3
HORN-8 board ($\times 8$):			
35,840 parallel (cores), 0.25 GHz	16	1,200	63

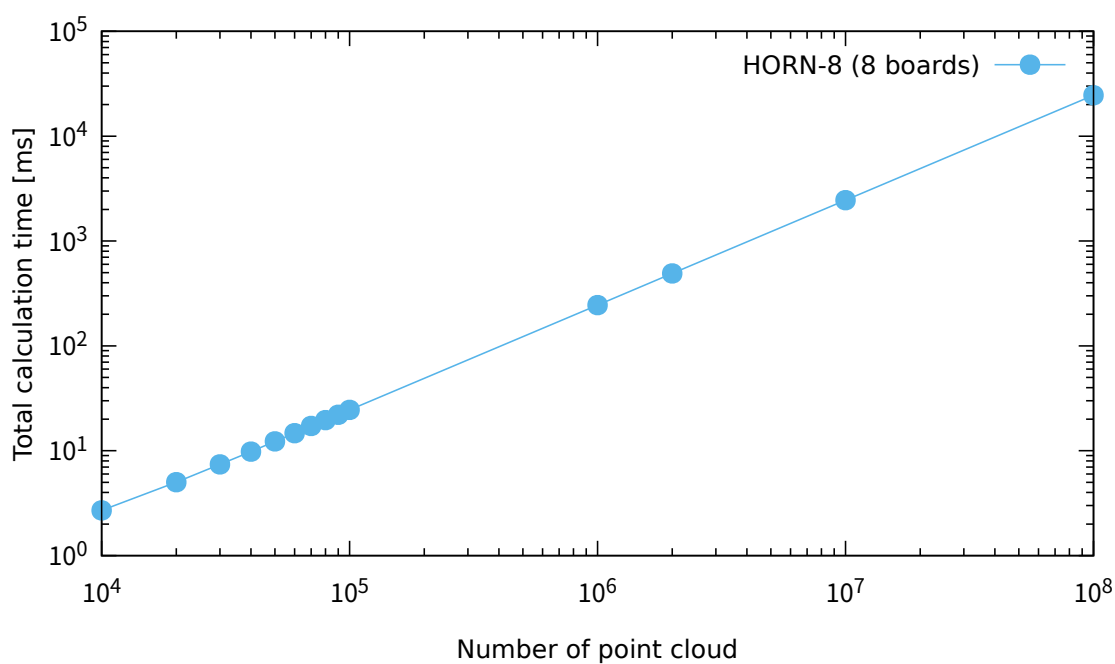


図 3.3: HORN-8 システムにおける点群数と計算時間の関係.

3.5 光学系

図3.4に再生光学系を示す. SLMは振幅変調型で, 画素ピッチ $6.4\ \mu\text{m}$, $1,920 \times 1,080$ 画素である. 再生照明光には $532\ \text{nm}$ の緑色レーザーを用いた. 再生像は SLM から $0.5\ \text{m}$ の位置に再生するように設定し, デジタルカメラで撮影した. カメラのシャッタ速度は人間の目の残像効果と同程度の $0.1\ \text{秒}$ ($10\ \text{fps}$) とした.

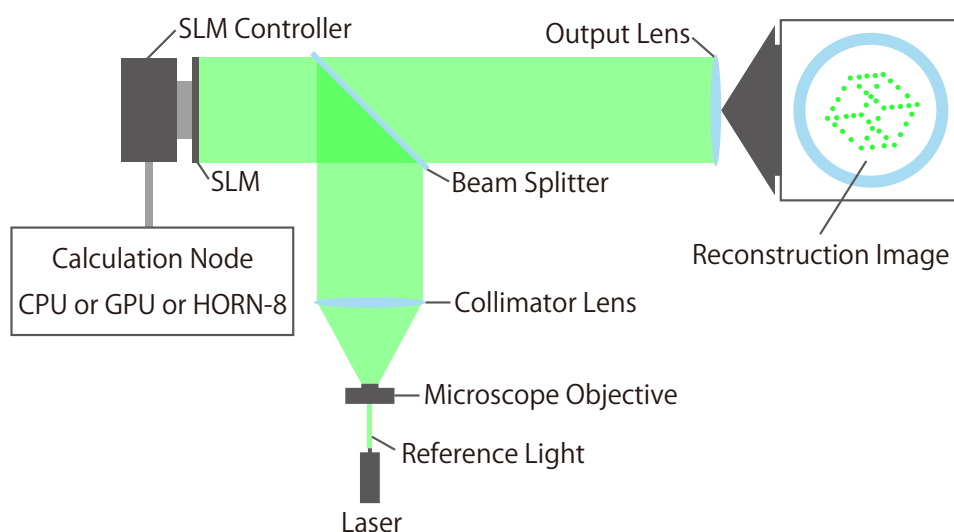


図 3.4: 時空間分割法を用いた再生光学系の様子.

3.6 再生像

再生像を図3.5に示した. (a)はHORN-8 クラスシステム, (b)はGPU, (c)はCPUの処理速度における再生像の違いを表している. 元画像は $390,000$ 点 (約 40 万点) の人工衛星である. 1 フレームごとに 6 分割して $65,000$ 点のデータ (フレーム) とし, ホログラム表示を行っている. つまり, フレーム数は 6 倍になる. (a) のHORN-8 クラスシステムでは, $65,000$ 点の点光源データから作成したCGHを $60\ \text{fps}$ で表示する. 6 フレームで $390,000$ 点の人工衛星を表現するため, 元画像に対するフレームレートは $10\ \text{fps}$ である. 各 $65,000$ 点の再生フレームが残像効果により合成され, 39 万点 ($65,000 \times 6$) の再生像のように観察されている. (b) のGPUでは, $65,000$ 点に対して $4.8\ \text{fps}$ なので, 元画像 ($390,000$ 点) に対しては $0.8\ \text{fps}$ 程度になる. (c) のCPUでは, $65,000$ 点に対して $0.052\ \text{fps}$ なので, 元画像 ($390,000$ 点) に対しては $0.001\ \text{fps}$ 程度である. (b), (c) とともに, 残像効果が発生するよりも再生速度が遅いため, 6 分割された $65,000$ 点の再生像がそのまま認識され, 画質が粗いことがわかる.

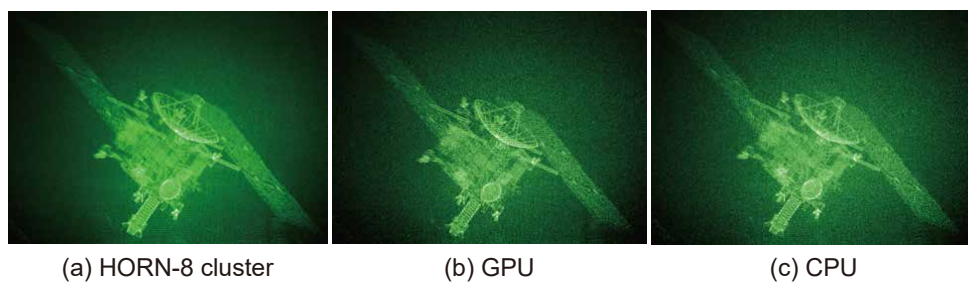


図 3.5: 計算速度による再生像の違い.

3.7 小括

本研究では、約 40 万点の点光源データを分割して再生可能なことを示し、専用計算機内のメモリの容量は少量でも問題ないことを明らかにした。少ないメモリであっても、1 フレームの計算にかかる時間が十分少なければ、時分割で再生することで、より大規模な点光源データも扱うことが可能である。また、粗い SLM に対しても高画質化が可能になる。これは、今後より高精細なデータを表現する場合、専用計算機内のメモリ量の問題よりも、いかに 1 フレームの計算にかかる時間を削減できるかが問題になるということを示している。電子ホログラフィ計算における専用計算機は、一般的な計算機と異なり、並列数の増加に応じて高速化するアーキテクチャであることが示されている [35]。今後、専用計算機内の計算コアを高密度、高集積化していくことにより、内蔵メモリの容量は少なくとも、1 フレームの計算にかかる時間を短縮することで、大規模な電子ホログラフィを扱えることを示唆するものである。

第4章 Zynq UltraScale+ MPSoC を用いた電子ホログラフィ専用 計算機

本章の目的は、電子ホログラフィ専用計算システム全体の小型化を達成することである。先行研究では、FPGA ボードをデスクトップ PC に接続した大型なシステムの実装が行われた [35, 36]。本章の研究では、System-on-a-Chip (SoC) を用いることで、デスクトップ PC + 計算アクセラータといった大型なシステムをチップレベルに小型化した。振幅型と位相型の Computer-Generated Holograms (CGH) を計算する小型な電子ホログラフィ専用計算機をそれぞれ開発した。また、ハンドモーショントラッカーを利用したインタラクティブシステムの開発を行い、将来的に電子ホログラフィを用いた Head Mounted Display (HMD) で利用可能なシステムの検討を行った。

4.1 Zynq UltraScale+ MPSoC について

Zynq UltraScale+ MPSoC (ZynqMP) は、Xilinx 社が提供する SoC である。FPGA チップ上に、組み込み機器でよく用いられる小電力・小型な ARM CPU を搭載する。FPGA に搭載された Look-up table をベースとして論理関数を提供するブロックを Soft Logic と呼ぶ。一方で、Look-up table を利用せず論理関数を提供するブロックを Hard Logic と呼ぶ。FPGA の開発では、特定の機能単位でまとめられた再利用可能な回路情報が Intellectual Property Core (IP コア) として提供されていることがある。CPU 機能を提供する IP コアも提供されており、従来より MicroBlaze [72] という IP コアが Xilinx 社から提供されている。MicroBlaze は、Soft Logic として提供されており、FPGA 上の LUT などのリソースを利用し実装される。そのため、動作周波数は低く、100 MHz 程度であった。一方で、ZynqMP は組み込み CPU を Hard Logic として搭載している。実装に LUT などを消費せず、専用回路で動作するため、動作周波数は 1.2 GHz を達成する。

ZynqMP を用いることで、Operating System (OS) を組み込み CPU 上で動作させ、C 言語などを用いたプログラムを用いて、実用的なアプリケーションを動かす

ことが可能である。FPGA に構築した専用計算回路を組み込み CPU から計算アクセラレータとして利用することで、デスクトップ PC であっても時間がかかる処理を小さなまま高速に処理することが可能となる。

4.2 組み込みシステム向け電子ホログラフィ専用計算機の構成

専用計算機システム開発のため、組み込み CPU・FPGA 間の通信回路の開発と専用計算回路の開発を行った。組み込み CPU から FPGA に構築した専用回路を利用する構成とし、CGH の計算フローを 2 つに分割し、組み込み CPU と FPGA で協調計算を行った (図 4.1)。式 (1.17) 中、 $\rho_j = 1/2\lambda|z_j|$ は除算を含む。一般的に、FPGA で除算を実行する場合、リソースの増加や動作周波数の低下につながる。 ρ_j の計算は事前に計算可能であり、組み込み CPU を用いて計算を行った。組み込み CPU で計算した ρ_j を FPGA 側へ送り、CGH の計算を行った。FPGA 側には CGH 専用計算回路を構築し、組み込み CPU から計算アクセラレータとして利用することで小型で高速なシステムを開発した。

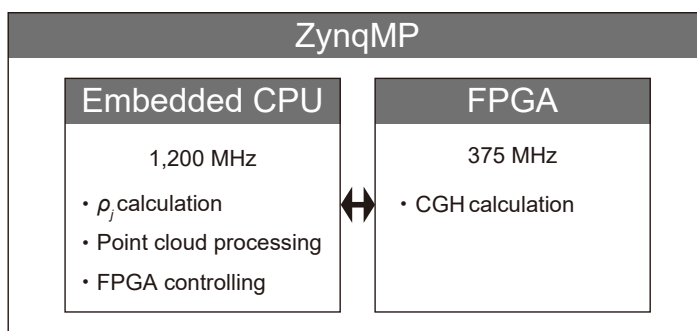


図 4.1: 専用計算システム内部の役割分担。

4.2.1 ターゲットボード

研究の初期段階として、Xilinx 社が提供する Zynq UltraScale+ MPSoC ZCU102 評価キット (以下、ZCU102 と呼ぶ) を用いた。ZCU102 は、Zynq UltraScale+ MPSoC ファミリの一つである ZU9EG-2FFB1156I を搭載している。ZCU102 の外観を図 4.2、搭載チップ内の FPGA 部分に関する仕様を表 4.1、組み込み CPU 環境については表 4.2 に示した。表 4.1、表 4.2 より、ZynqMP ではワンチップ内に組み込み CPU と FPGA が搭載されているが、それぞれ異なった動作周波数で動かしていることがわかる。組み込み CPU は 1,200 MHz で動作し、FPGA に構築した CGH 専用計算回

路は 375 MHz で動作する。表 4.1 中, Digital Signal Processor (DSP) は乗算や加減算などを行う信号処理に特化した Hard Logic である。本研究では, DSP を加算回路の一部を構成するのに利用し, Look-up Table や Flip Flop の使用数を削減した。

表 4.2 より, 組み込み CPU 上では, Ubuntu 18.04.3 をベースとして改良した Operating System (OS) が動作している。ZynqMP で OS を動作させるためには, ブートローダや Linux カーネル, システム動作に必要なファイル群である root file system (rootfs) といった OS の起動・実行に必要なファイルを作成する必要がある。Xilinx 社が提供する PetaLinux ツール [73] を利用することで, ブートローダや Linux カーネル, rootfs を生成し, 組み込み CPU で上で Linux が動作するシステムを構築することが可能である。しかし, PetaLinux ツールで生成したシステムは, パッケージの追加やドライバの開発において細かな調整が難しい。そのため, 本研究では Ubuntu 18.04.3 の rootfs を利用し, 独自にカスタマイズを行ったブートローダと Linux カーネルを組み込んだ OS を開発し, 利用した。本研究の本質ではないが, 開発した OS は, 動的な FPGA 再構成が可能であり, 一般的な Ubuntu OS を搭載した PC 同様, 自由にパッケージの追加が行える。専用計算機開発を効率的に行うことができるシステムである。

図 4.2 の外観に示すように, 利用した FPGA ボードは評価用ボードであり, 様々なインターフェースが付属しているため, 評価ボード自体のサイズは 25 cm × 25 cm である。一方で, 本稿のシステムはワンチップで実装が完結している。チップ自体のサイズは 3.5 cm × 3.5 cm であり, 最終的にチップのみを搭載した専用基板を開発することで HMD にも組み込み可能である。

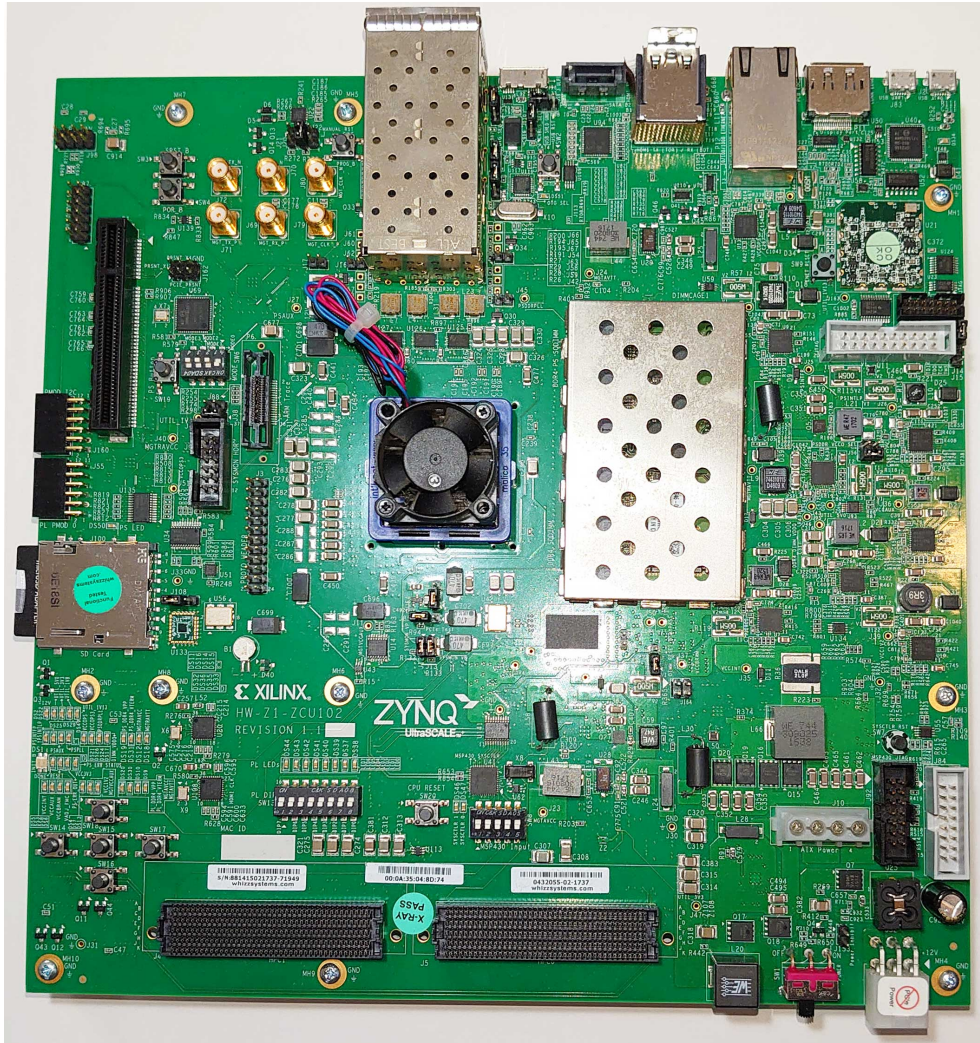


図 4.2: Zynq UltraScale+ MPSoC ZCU102 評価キットの外観. 黒色のファンの下に FPGA チップが搭載されている.

表 4.1: ZCU102 の論理回路リソース.

Resource	Available
Look-up Table	274,080
Flip Flop	548,160
Block RAM	32.1 Mb
DSP	2,520

表 4.2: ZCU102 の CPU 環境.

CPU	Quad-core ARM Cortex-A53
Operating frequency	1,200 MHz
OS	Ubuntu 18.04.3 (Linux Kernel 4.19.0)

4.3 組み込み CPU ・ FPGA 間通信

ZynqMP では、組み込み CPU と FPGA は Advanced eXtensible Interface (AXI) という通信プロトコルを利用するバス線により接続されている。組み込み CPU と FPGA 間の通信には、AXI に対応した通信回路を使用する必要がある。AXI 通信回路として、Xilinx 社から AXI Direct Memory Access (AXI DMA) [74] という IP コアが提供されている。本章の電子ホログラフィ専用計算システムでは、FPGA 側に構築した数千の並列計算コアが生成する数千のデータを効率的に転送する必要がある。AXI DMA の仕様上、電子ホログラフィ専用計算機では通信のボトルネックが発生し、FPGA 側に構築する計算回路の並列数を増やしても律速する問題点がある。本研究では、電子ホログラフィ専用計算機実装のため、まず専用通信回路の開発を行った。

4.3.1 AXI プロトコルの基礎

AXI は、ARM 社が制定するモジュール間通信プロトコルである。AXI-Full, AXI-Lite, AXI-Stream といった 3 つの規格がある。AXI-Full は、AXI 全ての規格を利用でき、データの送受信を高速に行うことができる通信形式である。先頭アドレスを一度指定し、その後はアドレスの指定なしにデータの送受信を行い、通信の高速化を図るなど高速な通信手法に対応したプロトコルである。AXI Lite は、AXI-Full を簡略化した軽量通信プロトコルである。主に、モジュールのコントロール用のプロトコルであり、アドレスを指定し、1 つデータを送受信するといったプロトコルとなっている。AXI-Stream は、常に一方向のモジュールへデータを受け渡す高速かつ軽量なプロトコルである。制御信号の通信には AXI Lite を利用し、大規模なデータの通信には AXI-Full や AXI-Stream を利用するといった使い分けを行う。

AXI プロトコルの基本としては、モジュール間の一対一関係を考えればよい。例として、A モジュールと B モジュール間の通信模式図を図 4.3 に、通信波形を図 4.4 に示した。DATA はデータを送るための信号線である。32 bit, 64 bit, 128 bit 幅などの AXI で決められたデータ幅を利用可能である。ここでのデータは数値データだけでなく、アドレスといった値も含む。VALID と READY は、1 bit の制御線であ

る。VALIDは送信側が有効なデータを提示しているときのみ1を出力する。READYは受信側がDATAを受信可能な場合、1を出力する。VALIDとREADYが両方1となったとき、転送が完了する(図4.4中、斜線部)。一方がVALIDで有効なデータを提示していることを示し、もう一方がREADYで受信可能であることを示す通信形式をここではVALID-READY通信と呼ぶ。また、VALID-READY通信を行うひとまとまりの信号線をチャンネルと呼ぶ。

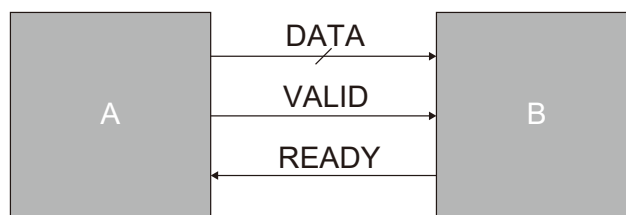


図 4.3: AXI の通信モード (1 Channel).

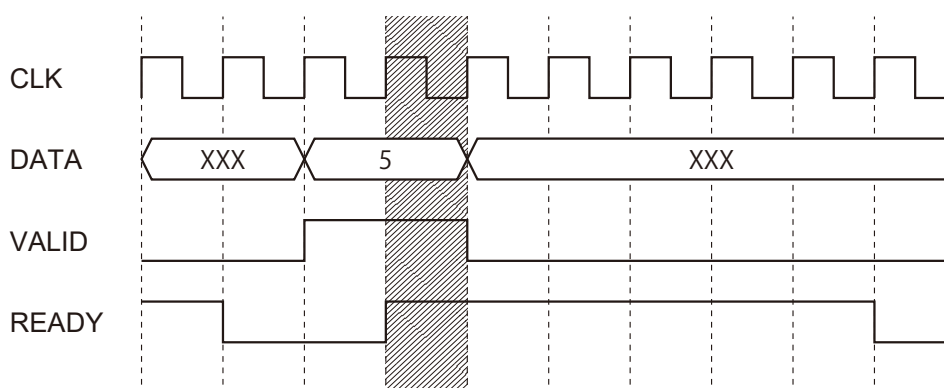


図 4.4: AXI の通信波形 (1 Channel).

図4.3のAモジュールとBモジュールの例では、データは常にAモジュールからBモジュールへと送られていた。実際のAXI-FullとAXI-Liteでは、複数のチャンネルで送受信を行い、データ送信用のアドレスチャンネル(AW Channel)、データ送信用チャンネル(W Channel)、受信応答用のチャンネル(B Channel)、データ受信用のアドレスチャンネル(AR Channel)、データ受信用チャンネル(R Channel)を利用する(図4.5)。AXIプロトコルは、VALID-READY通信を基本として、複数チャンネルにより通信を行う通信プロトコルである。

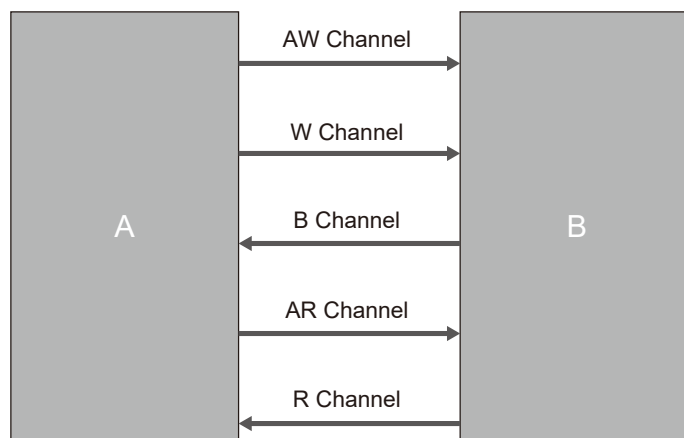


図 4.5: AXI の通信模式図 (全体).

データの送受信に関しては常に一方のモジュールが決定権を持つ (図 4.5 の例では A モジュールが相当). 決定権を持つモジュールがアドレスチャンネルを利用し, データの書き込み, もしくはデータを読み取るアドレスの指定を行う. もう一方のモジュール (図 4.5 の例では B モジュールが相当) は, 指定されたアドレスに対応するデータの送受信を行う. このように AXI-Full, AXI-Lite に関しては, データ自体は双方向を行き来する. 一方, AXI-Stream ではアドレスを指定するチャンネルが無い, データは常に一方向へと送信される.

4.3.2 高速な AXI 通信方法

AXI-Full では, 高速な通信のため, バースト転送とオーバーラップ転送といった転送方式が利用可能である. バースト転送では, 一度のみ送受信するアドレスの指定を行い, そのアドレスから連続した位置にあるデータをそのまま続けて送受信する通信方式である. 例として, 図 4.6 に A モジュールから B モジュールへデータを書き込むために利用する AW Channel と W Channel のデータの流れを図示した. 通信はクロック信号 (CLK) に同期して実行される. XXX は不定値を示す. Address[0] により, 一度アドレスを指定し, その後 N ワードのデータを連続して送ることが可能である. AXI では一度に連続して送れるデータ数は 256 ワードである. バースト転送を利用しない場合 (図 4.7) は, アドレスを指定し, 1 つデータを送るといった通信をデータ数回繰り返す必要がある. 図 4.7 では, 一度アドレスを指定し, データを送信した後, 直ちに次のアドレスを指定し, データを送る理想的な模式図を示した. 実際にはデータ送信後に次のアドレスを指定可能になるまで, いくらか待ち時間が必要である. 一つのアドレスに対し一つのデータを送る方式は, AXI Lite に相当する動作である. AXI-Full では, オーバラップ転送を利用することで, 多くの

データを一度のアドレス指定で送受信でき、その分高速に通信が可能である。オーバーラップ転送は、アドレスの送信とデータの送受信を並行して行う通信方式である(図4.8)。アドレスとデータを並行して送受信できるため、その分高速に通信が可能になる。図4.8の例では、オーバーラップ転送のみ適用し、バースト転送は利用していない。オーバーラップ転送とバースト転送の両方を組み合わせることができ、組み合わせることでより高速な通信が可能となる。

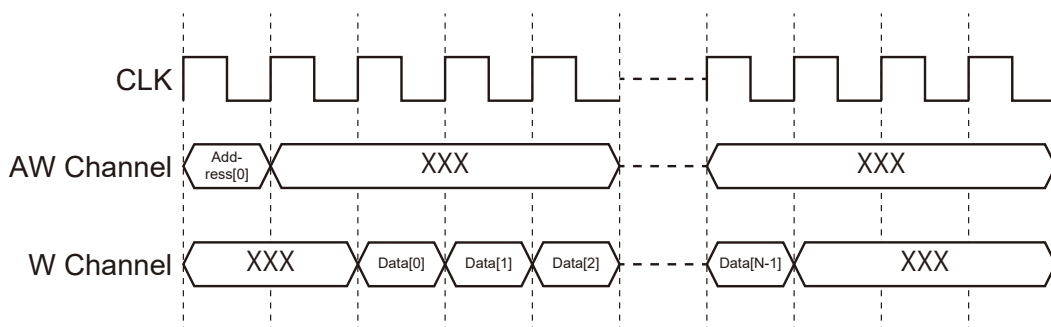


図 4.6: バースト転送のイメージ。

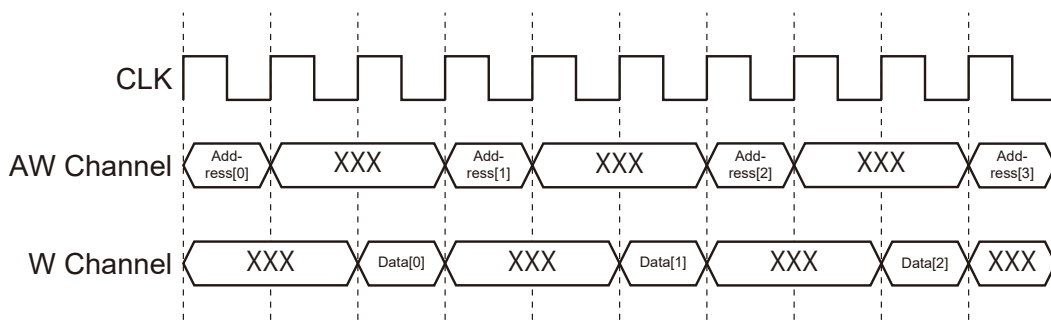


図 4.7: バースト転送とオーバーラップ転送を利用しない場合の転送イメージ。

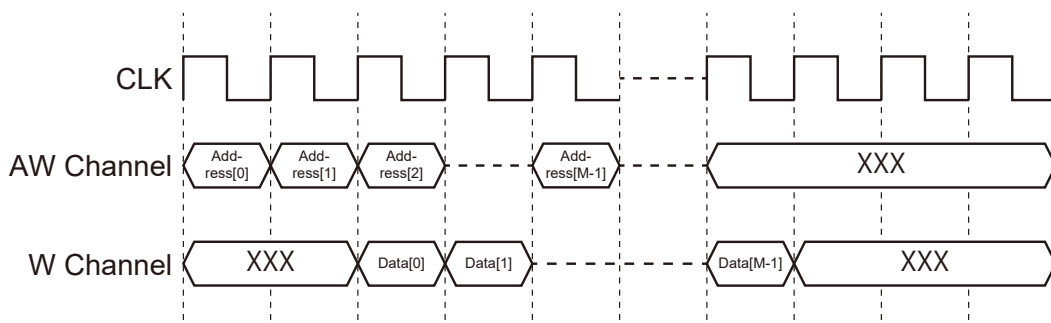


図 4.8: オーバーラップ転送のイメージ。

4.3.3 AXI 通信回路の構成

ZynqMP では、組み込み CPU と FPGA 通信に AXI Lite と AXI-Full が利用可能である。AXI Lite を利用した通信回路は、組み込み CPU 側から FPGA 側のモジュールを制御することができる。AXI Lite では、バースト転送やオーバーラップ転送が利用できないため、低速な通信となる。AXI-Full を利用した通信回路は、FPGA 側から外部のメインメモリ (Dynamic Random Access Memory: DRAM 上のデータ) を直接制御することができる。CPU を介さず周辺機器と通信する方式である Direct Memory Access (DMA) 転送が可能であり、組み込み CPU による性能の影響を受けない高速な通信が可能である。AXI の通信自体もバースト転送やオーバーラップ転送が利用可能であり、AXI Lite と比べ、大量のデータを高速に送受信可能である。

電子ホログラフィ専用計算機向けには、組み込み CPU を中心として FPGA 側の専用計算機を制御する必要がある。組み込み CPU から専用計算機へ向けて、点群数などのパラメータの送信、計算開始や終了用の制御レジスタの読み書きが必要であるため、AXI Lite を用いた通信を行った。一方で、AXI Lite だけでは、通信速度が遅く、点群データや計算結果の CGH データといった大規模データ送受信にかかる時間がボトルネックとなる。点群データと CGH データの送受信は、AXI-Full を使用した DMA 通信により、高速に読み書きを行う構成とした (図 4.9)。ただし、AXI-Full ではバースト転送とオーバーラップ転送が利用できるが、AXI を利用した通信回路開発の初期段階として、バースト転送のみに対応した通信回路を開発した。FPGA 内部に AXI Lite と AXI-Full に対応した通信回路を実装し、専用計算機と組み込み CPU を通信可能にした。

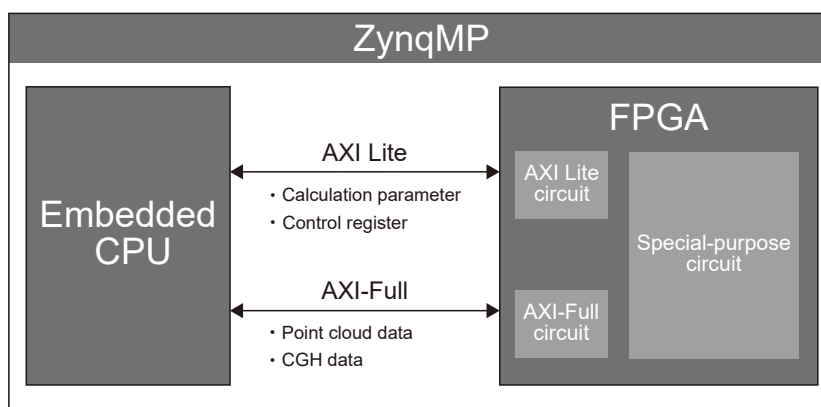


図 4.9: 電子ホログラフィ専用計算機向け AXI 通信回路の様子。

4.3.4 AXI Lite 通信回路の実装

組み込み CPU から FPGA に構築する専用計算機を制御するために、AXI Lite プロトコルに対応した通信回路を開発した。AXI Lite を用いて、点群数などのパラメータの送信、計算開始や終了用の制御レジスタの読み書きを行う。また、後述の AXI-Full 通信回路を制御する目的でも利用する。AXI Lite 通信回路は、パラメータ等の少量のデータ通信を目的とし、データ幅は 32 bit、アドレス空間は 2^{10} Byte とした。開発した AXI Lite 通信回路のアーキテクチャを図 4.10 に示した。

電子ホログラフィ専用計算機向けの AXI Lite 通信回路は、組み込み CPU からの書き込み処理に応答する write responder と読み取り処理に응答する read responder、各制御信号のレジスタ群から構成した。組み込み CPU から FPGA への書き込み応答のため、write responder には、AW Channel と W Channel と B Channel を接続した。組み込み CPU から FPGA への読み取り応答のため、read responder には、AR Channel と R Channel を接続した。write responder と read responder は、ステートマシンで構成され、AXI Lite の各信号線の変化に合わせて、データを制御信号用のレジスタへと割り振る。レジスタはそれぞれ後述の AXI-Full 通信回路、もしくは CGH 専用計算回路へ接続されている。write responder の状態遷移図を WRITE ステートマシンとして図 4.11 に、read responder の状態遷移図を READ ステートマシンとして図 4.12 に示した。

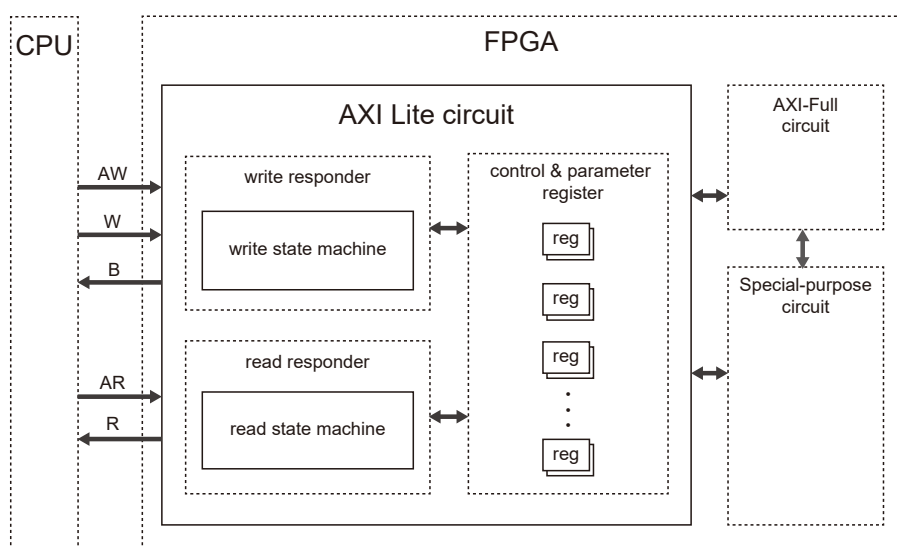


図 4.10: AXI Lite 通信回路のブロック図.

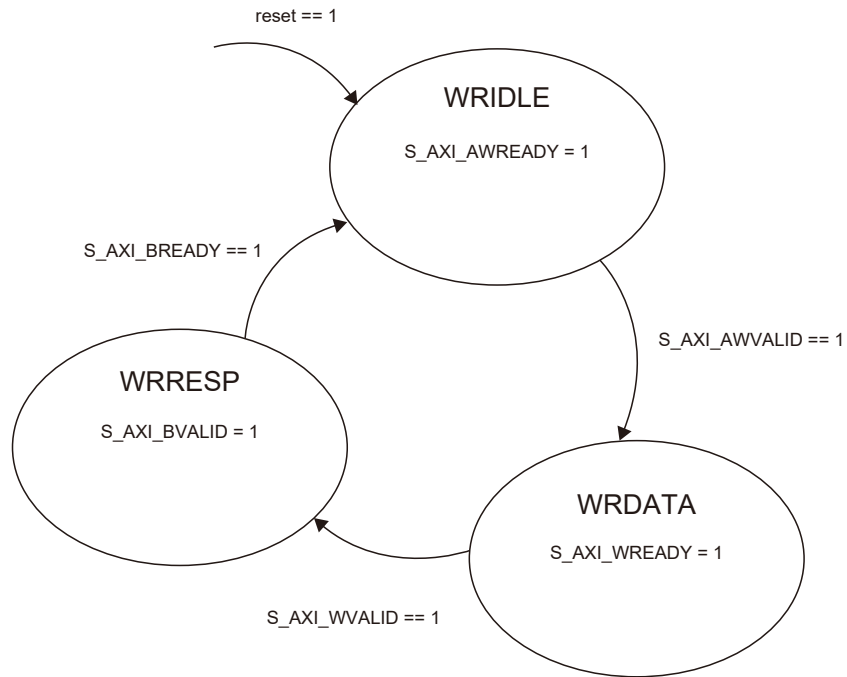


図 4.11: write responder の状態遷移図.

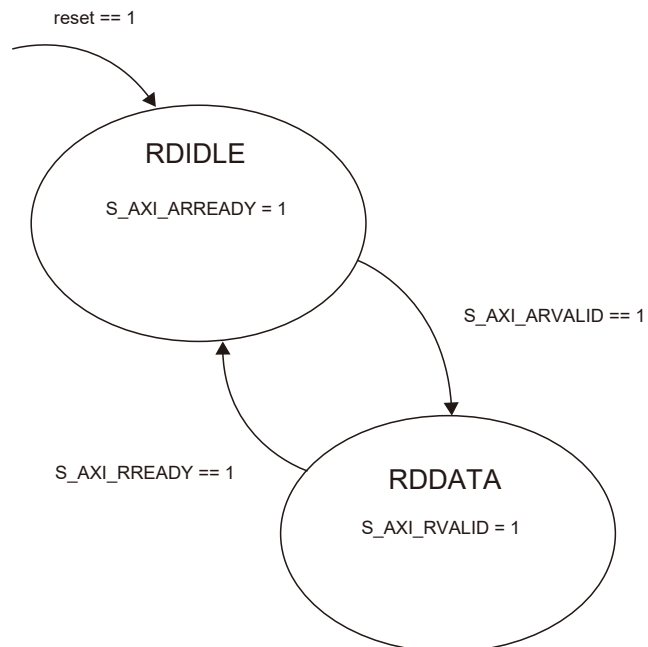


図 4.12: read responder の状態遷移図.

図4.11に示したように、WRITE ステートマシンは、書き込み待ちステートである WRIDLE、書き込み実行を行う WRDATA、書き込み応答を行う WRRESP ステートで構成した。回路全体は、reset 信号により初期化される。reset 信号は負論理を利用しており、リセット (reset = 0) 後、reset = 1 で WRIDLE ステートから実行する。WRIDLE ステートでは、アドレスを受信可能であることを示す信号 S_AXI_AWREADY を 1 にし、入力を待つ。アドレスが有効な値を出力していることを示す信号 S_AXI_AWVALID が 1 になったとき、受信可能信号 S_AXI_AWREADY を 0 にし、WRDATA ステートへ切り替わる。WRDATA ステートでは、データが受信可能であることを示す S_AXI_WREADY を 1 にする。データが有効な値を示していることを示す信号 S_AXI_WVALID が 1 となったとき、データ受信可能信号 S_AXI_WREADY を 0 に戻し、WRRESP ステートへ遷移する。WRRESP では読み取り成功応答を S_AXI_BVALID を 1 にすることで返す。書き込みの成否を表す S_AXI_BRESP は、常に成功の 0 を返す。読み取り成功応答を組み込み CPU が受け取ったことを示す S_AXI_BREADY が 1 となったとき、S_AXI_BVALID を 0 に戻し、WRIDLE ステートへと戻る。

WRITE ステートマシンにより、組み込み CPU からデータを FPGA 内部へ取り込む。指定されたアドレスを元にして、各制御信号用のレジスタへとデータの書き込みを行う。各制御信号用レジスタは、後述の AXI-Full 通信回路もしくは、専用計算回路へと接続されている。

READ ステートマシンは、読み取り待ちステートである RDIDLE と読み取り実行を行う RDDATA ステートで構成した。WRITE ステートマシン同様、初期化後、RDIDLE ステートから実行する。RDIDLE ステートでは、アドレスを受信可能であることを示す S_AXI_ARREADY を 1 にし、入力を待つ。有効なアドレスを出力していることを示す信号 S_AXI_ARVALID が 1 になったとき、アドレス受信可能信号 S_AXI_ARREADY を 0 に戻し、RDDATA ステートへ遷移する。RDDATA ステートでは、有効なデータを出力していることを示す信号である S_AXI_RVALID を 1 にする。S_AXI_RREADY が 1 になったとき、S_AXI_RVALID を 0 に戻し、RDIDLE ステートへと戻る。

READ ステートマシンにより、組み込み CPU から指定されたアドレスに対応する FPGA 内部のデータを返信する。返信するデータは、WRITE ステートマシンにより取り込んだデータもしくは、AXI-Full 通信回路や専用計算回路から入力される信号である。

本研究では読み書きの処理 (WRITE ステートマシンと READ ステートマシン) がそれぞれ独立した実装である。この場合、同じレジスタに対し読み書きが同時に発生した場合、レジスタの値は不定値となる。一方で、後述する電子ホログラフィ専用計算機では、同時に同じレジスタにアクセスすることはなく、シンプルな実装が

可能な独立した実装を選択した。

組み込み CPU から見た FPGA 内部のアドレスとレジスタの名前について表 4.3 に示した。Local signal の [] は、ビットの位置もしくは範囲を示す。0–20 までのアドレスは DMA 関連アドレスである。AXI-Full 通信回路に接続されている。40–68 は CGH 専用計算回路関連アドレスである。CGH 専用計算回路へ接続されている。Local signal に”_reg” がついたものが書き込み可能なレジスタである。他は読み取り専用である。reserved は利用できないことを示す。

DMA 関連の w_start_reg は書き込み DMA の開始をコントロールするレジスタである。w_ready は書き込みステートマシンがアイドル状態であることを示す。r_ready は読み込みステートマシンがアイドル状態であることを示す。w_addr_reg は FPGA へ転送するデータが保存された物理メモリ (DRAM) のアドレスを指定する。w_word_num_reg はデータ数 (Byte 単位) を指定する。r_start_reg は読み込み DMA の開始をコントロールするレジスタである。r_addr_reg は FPGA から転送するデータを保存する物理メモリアドレスを指定する。r_word_num_reg はデータ数 (Byte 単位) を指定する。

CGH 専用計算回路関連の clc_ready は計算回路がアイドル状態であることを示す。clc_start_reg は、計算開始をコントロールするレジスタである。x_px_max_reg は、計算する CGH の横幅を指定する。y_px_max_reg は、計算する CGH の高さを指定する。object_num_reg は、計算する点群数を指定する。

本章の DMA 通信に利用する物理メモリアドレスは連続になっている必要がある。連続したアドレス確保のため、dma_alloc_coherent() という Linux カーネル Application Programming Interface (API) を利用し、固定の DMA バッファを確保した。ただし、ZynqMP 内部の組み込み CPU は 64 bit システムである。そのままでは確保される物理アドレスも 64 bit の範囲となる。一方で実装したシステムは表 4.3 に示したように 32 bit までしか指定できない。システムにより確保される DMA バッファのアドレスが 32 bit の範囲になるように dma_set_mask() API を利用し、制限することで対応した。

表 4.3: アドレスマップ (ZCU102).

Address	Register Name	Local signal
0	DMA_FLAG	[2]:r_ready,[1]:w_ready,[0]:w_start_reg
4	DMA_W_ADDR	[31:0]:w_addr_reg
8	DMA_W_WORD_NUM	[31:0]:w_word_num_reg
12	DMA_R_START	[0]:r_start_reg
16	DMA_R_ADDR	[31:0]:r_addr_reg
20	DMA_R_WORD_NUM	[31:0]:r_word_num_reg
24 36	NONE	reserved
40	CLC_FLAG	[1]:clc_ready,[0]:clc_start_reg
44 52	NONE	reserved
56	X_PX_MAX	[31:0]:xpx_max_reg
60	NONE	reserved
64	Y_PX_MAX	[31:0]:ypx_max_reg
68	OBJECT_LEN	[31:0]:object_num_reg

4.3.5 AXI-Full 通信回路の実装

点群データや計算結果の CGH データといったひとまとまりのデータ送受信を高速に行うために、AXI-Full プロトコルに対応した通信回路の開発を行った。開発した AXI-Full 通信回路は、FPGA の専用回路で計算した CGH データを組み込み CPU のメインメモリ (DRAM) へ書き込む際に、専用計算回路と非同期に転送可能な仕様とした。専用回路の計算実行と同時に計算結果の転送を行うことが可能であり、データ通信時間の隠蔽を行うことが可能である。開発した AXI-Full 通信回路のアーキテクチャを図 4.13 に示した。

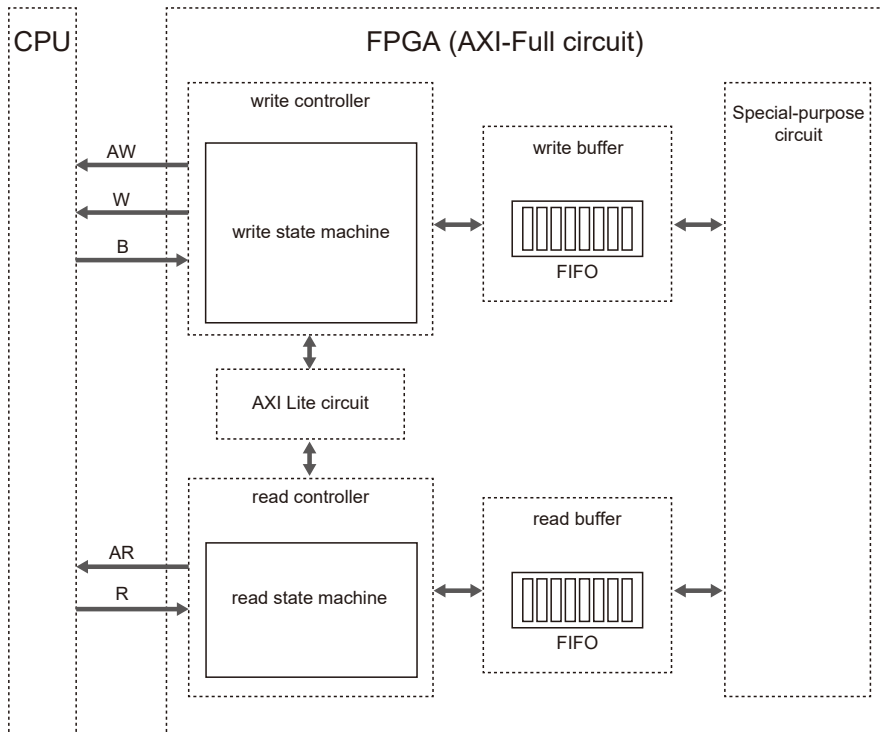


図 4.13: AXI-Full 通信回路のブロック図.

電子ホログラフィ専用計算機向けの AXI-Full 通信回路は、FPGA 内部のデータを外部のメインメモリへと転送する write controller とメインメモリにあるデータを FPGA 内部へと転送する read controller、データを一時保存するバッファから構成した。バッファは First In First Out (FIFO) で構成し、データのバッファリングを行う。バッファリングを行うことで、バースト転送とデータの非同期通信が可能になる。図 4.14 にバッファ FIFO の入出力の様子を図示した。

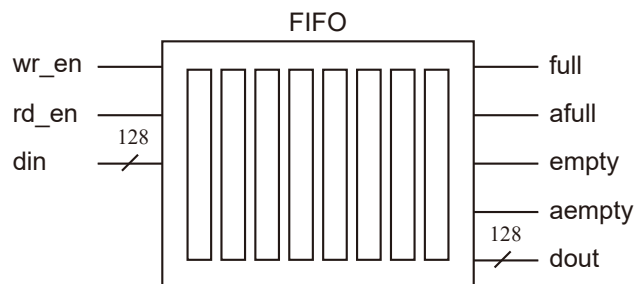


図 4.14: バッファ FIFO の入出力端子の様子.

FPGA からメインメモリへデータを転送するため、write controller には、AW Chan-

nel と W Channel と B チャンネルを接続した。メインメモリからデータを FPGA 内部へ転送するため、read controller には、AR Channel と R Channel を接続した。write controller と read controller は、ステートマシンで構成され、前述の AXI Lite から入力される制御信号と AXI-Full の各信号線の変化に合わせて、データの転送を実行する。AXI Lite 通信回路は、通信の制御は組み込み CPU にあり、要求に応答するモジュール (responder) であった。一方、AXI-Full 通信回路は、通信の制御権を持っており (controller)、CPU の介在なしに、外部のメインメモリと直接データのやりとりを行うモジュールである。

FPGA 内部のデータを外部のメインメモリへ転送する write controller の状態遷移図を WRITE ステートマシンとして図 4.15 に図示した。AXI 通信の流れ自体は AXI Lite 用の回路と同じである。ただし、バースト転送対応のため、ステートが複雑になっている。まずアドレスのやりとりを行い、内部の FIFO の状態に応じて、できる限りバースト転送でデータを転送する設計となっている。

初期化後、ステートは W_IDLE から実行する。W_IDLE では、AXI Lite 回路から入力される値である転送するデータ数 (Byte) を内部の回路に取り込む。同じく、AXI Lite 回路から入力される転送開始制御信号 W_START が 1 になったとき、W_PROC_START へ遷移する。FPGA 内部から外部のメインメモリへ転送するデータが記録されている FIFO の空き状況を表す aempty シグナルと、reg_w_word_num の数値を確認し、W_ADDR_START へ遷移する。reg_w_word_num の数値がバースト転送可能な値以上残っているかを確認する。ここではデータ幅が 128 bit であることを仮定しているため、実際に比較する値は 128×256 となる。一度にバースト転送可能な値よりも多い場合は、バースト転送数を記録する reg_w_burst_len には、最大値の 256 を記録する。reg_w_word_num の数値が一度にバースト転送可能な値以下の場合、残りの数を記録し、転送の終了用フラグ reg_w_last に 1 を記録する。また、書き込み先のアドレスを転送する。

ステートは W_DATA_WAIT へと遷移し、M_AXI_AWREADY が 1 になり、アドレスが受信されたとき、次のステートへ遷移する。W_DATA_PROC では、一つの転送が成功するたびに、reg_w_burst_len をカウントダウンし、0 になったとき、次のステートへ遷移する。W_WAIT では、書き込み応答である M_AXI_BVALID が 1 になったとき、次のステートへ遷移する。転送の終了用フラグ reg_w_last の値によって、転送処理を継続する場合は W_PROC_START へ遷移し、終了する場合は W_IDLE へ戻る。このとき、厳密な AXI トランザクションでは、書き込み応答として、書き込みエラーが返ってくることが考えられる。しかし、チップ内部の転送であり、転送エラーは想定が難しく、常に成功を仮定した。研究用途としてのシンプルな実装のため、転送に失敗した場合の処理は実装しなかった。

る。このとき、転送の終了用フラグ `reg_r_last` の値によって、転送処理を継続する場合は `R_ADDR_WAIT` へ遷移し、終了する場合は `R_IDLE` へ戻る。書き込み処理と同様、研究用途としてのシンプルな実装のため、常に成功を仮定した転送回路を実装した。

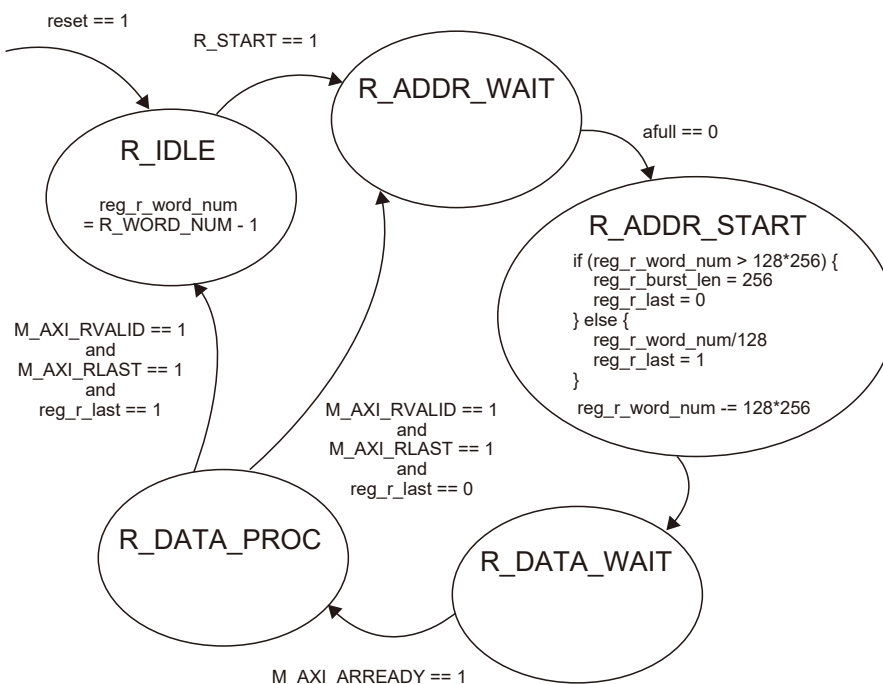


図 4.16: read controller の状態遷移図.

本章で開発した AXI-Full 通信回路は、組み込み CPU からアドレスを指定することができないストリーム型の通信回路である。本研究では、組み込み CPU へ CGH の計算結果を送信する場合、計算回路で生成されたデータをそのまま送信すれば良いため、アドレスが指定できなくとも問題ない。一方で、組み込み CPU から点群データを受信する場合は、FPGA 内部 (計算回路内部) のカウンタによりアドレスを復元することで Block RAM に保存可能にした。

4.4 振幅型電子ホログラフィ専用計算回路

漸化式法を利用した CGH 計算回路を FPGA に実装した。先行研究 [35] では 1 チップ 640 並列 (漸化式の適用画素数が 640 画素間隔) を前提とした設計が行われていた。一方で本章で利用する FPGA では、1 チップで数千~数万並列実装可能である。漸化式法を横方向に適用する場合、適用可能な画素数の最大数は CGH の横サイズと

なる。従って、1チップで複数ラインの漸化式計算を計算可能にするアーキテクチャの設計を行う必要があった。本章では、先行研究 [35] をベースとした計算精度の再検証と大規模FPGAでさらなる並列計算を可能とするアーキテクチャの設計を行った。なお、ここで検討した計算精度および回路アーキテクチャは、後述する位相型のシステムと5章のU250を用いたシステムでも基本は同じである。

4.4.1 振幅型の計算精度設定

CPUでは浮動小数点数が良く用いられる。浮動小数点数は指数表現を利用し、数値を表現する手法である。IEEE754規格などでデータフォーマットは決められている。大きな値を扱える一方、計算する際に指数の計算が必要となる。一方で、FPGAで数値計算する際には固定小数点数が良く用いられる。固定小数点数ではユーザが小数点の位置を任意に決定する。浮動小数点数と比べて扱える値の範囲は小さいが、計算の際に指数の計算は必要無く、単純な回路で計算可能である。同じデータ幅どうしで計算した場合、浮動小数点数はより大きな値を扱うことができるが、固定小数点数の方が精度は高くなる。

浮動小数点数では32 bitもしくは64 bitのデータ幅がよく利用される。FPGAでは任意のデータ幅を利用することが可能である。少ないデータ幅の方が演算器をより少ないリソース(ゲート数もしくはトランジスタ数)で作ることができ、回路全体として並列数を増やすことが可能である。

電子ホログラフィ計算を固定小数点数で計算する場合を考える。式(1.17)より、 x_a, x_j, y_a, y_j は整数値である。扱いたい数値が整数値の場合、固定小数点数の表現は簡単である。データ幅は変数が得る値の最小値と最大値をデータ幅とする。 (x_a, y_a) はCGH面上の座標、 x_j, y_j は点群の座標である。1,920×1,080画素のSLMを使うとき、一番大きな値(1,920)の入力に必要なデータ幅を考慮し、必要なデータ幅は符号無しで11 bitである。ここでは、将来的な拡張を見越して8Kサイズも計算可能なように余裕をもたせ、 x_a, x_j, y_a, y_j のデータ幅は符号あり14 bit入力とした。

図4.17に整数の固定小数点数のデータ幅と計算による小数点の位置を図示した。整数同士の固定小数点数の加減算では小数点の移動は発生せず、そのまま計算可能である。桁上がりを考慮して、加算では1bit、乗算ではデータ幅の和だけデータ幅を拡張し、各演算に必要なデータ幅が決まる。

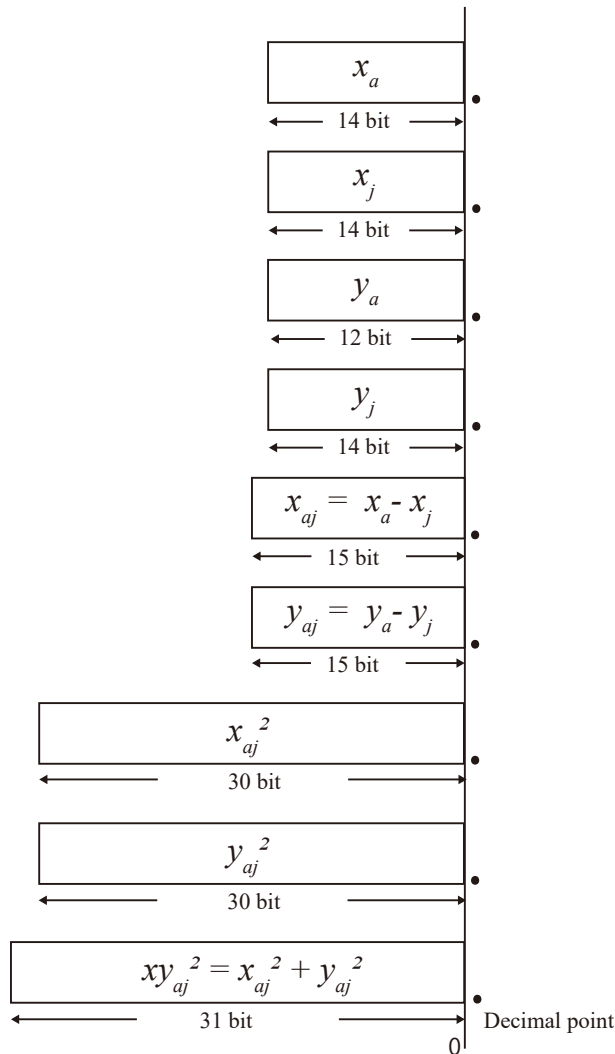


図 4.17: 電子ホログラフィ計算における整数値の固定小数点数表現.

回路への入力として、CGHの座標値である x_a, y_a と点群データである x_j, y_j, ρ_j を入力としてCGH計算を行う。ここで、式 (1.17) の $\rho_j = 1/2\lambda|z_j|$ は小数となる。また、この計算は除算を含む。一般的にFPGAで除算回路を実装する場合、リソースの増加や動作周波数の低下につながる。 ρ_j の計算は事前に計算可能であり、CPUを使用して計算を行い、 x_j, y_j, ρ_j として点群データを転送する仕様とした。

小数の2進数における固定小数点数での表現は、それぞれの桁が2の冪乗単位の重みを持つと考える。整数部はそれぞれ2の0乗、2の1乗と正の数値を指数部に持つ。同じく、小数部は、2の-1乗、2の-2乗といったように負の数値が指数部となった重みを持つ 図 4.18.

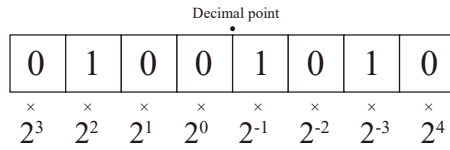


図 4.18: 固定小数点数表現.

先行研究 [35] では, ρ_j は 32 bit シフトした値を入力していた. また, 漸化式法における θ_j , Δ_{aj} , Γ_{aj} は 21 bit 幅に設定されていた. 本研究でも ρ_j は同様に 32 bit とした. 一方で, 漸化式を先行研究よりも多く適用するため θ_j , Δ_{aj} , Γ_{aj} は 22 bit 幅に設定した. 図 4.19 に小数の固定小数点数のデータ幅と乗算により小数点移動する様子を図示した.

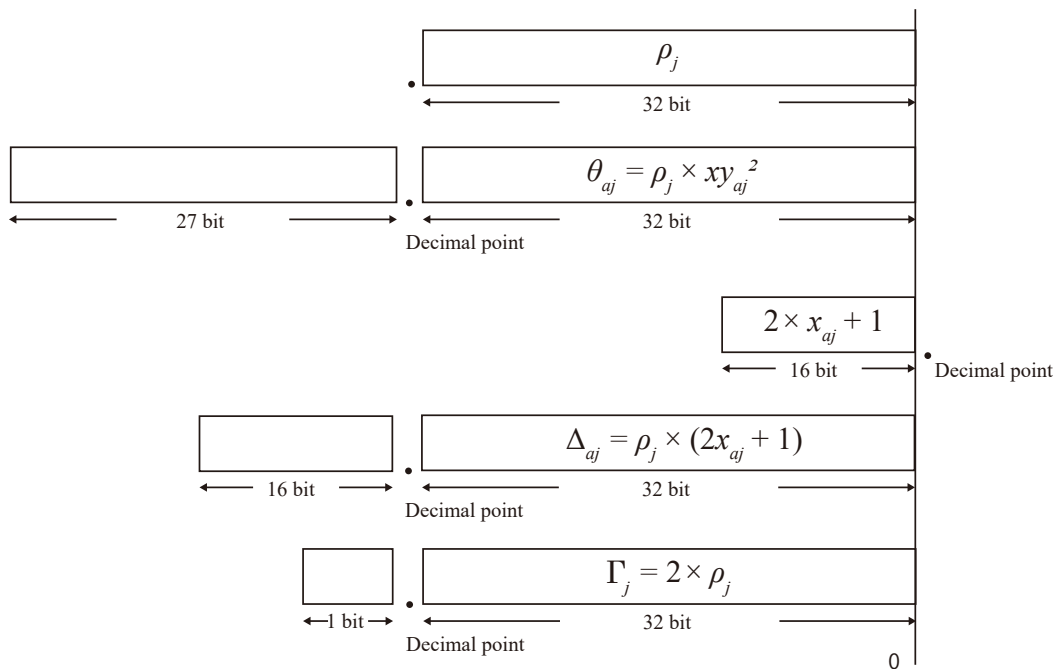


図 4.19: 電子ホログラフィ計算における小数を含む固定小数点数表現.

式 (1.18) の余弦計算には事前に計算結果をメモリに格納し, アドレス参照により高速に値を計算する Look-Up Table (LUT) 法を利用した. 振幅型 CGH の計算に必要な余弦 LUT の精度は先行研究 [35] で調査されており, ここでも 6 bit 入力, 6 bit 出力の LUT を利用した.

CGH 計算の精度に問題が無いことをシミュレーションベースで検証した結果を示す. 32 bit float 精度で計算した CGH と比較した CGH 同士の平均 2 乗誤差 (Mean

Squared Error: MSE) と、CGH から再生計算した再生像を画像評価指標の一種である PSNR を指標として比較した結果を図 4.20 に示した。SSIM との比較を図 4.21 に示した。65,000 点の点群から CGH を計算し、再生シミュレーション条件は、画素ピッチ $6.4\ \mu\text{m}$ 、 $1,920 \times 1,080$ 画素、再生照明光波 $532\ \text{nm}$ 、再生距離 $0.5\ \text{m}$ である。

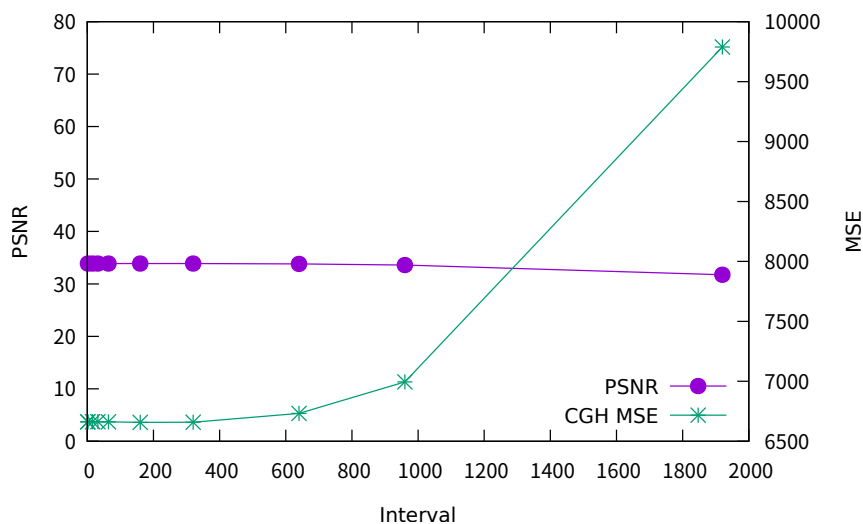


図 4.20: 再生像の PSNR と CGH の MSE 比較 (振幅型).

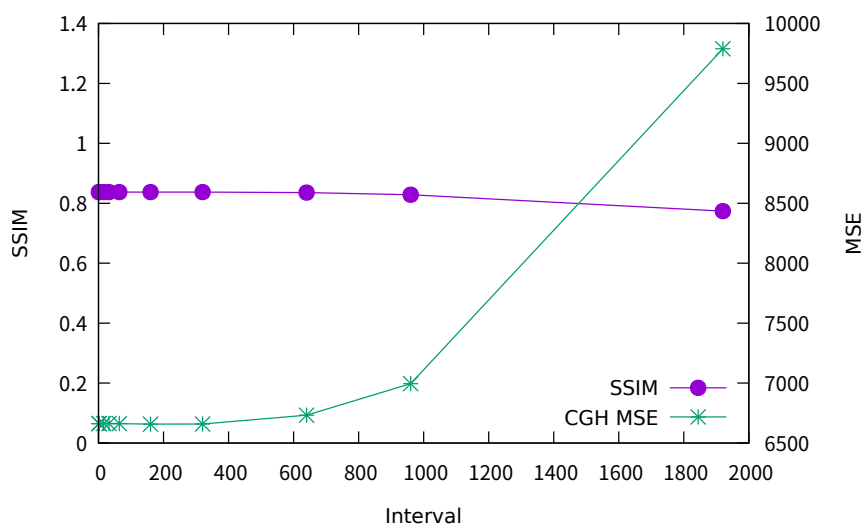
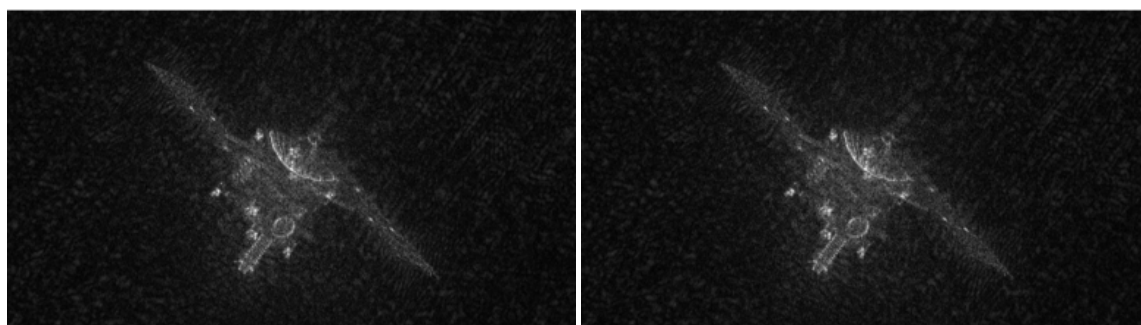


図 4.21: 再生像の SSIM と CGH の MSE 比較 (振幅型).

図 4.20, 図 4.21 では、漸化式の適用間隔と各値の変化を示した。漸化式の適用間隔が大きくなればるほど、CGH 自体の誤差は増えていっていることがわかる。一方で、再生像の画質はそれほど劣化しておらず、1,920 画素適用時、PSNR は $34.8\ \text{dB}$ 、SSIM は 0.842 と、横すべての画素に関して漸化式法を適用しても問題ないことがわ

かる。これはCGHの冗長性によるものであると考えられる。CGH上のそれぞれの画素値にはすべての点群の情報が圧縮されて保存されていると考えることができる。そのため、一般的にホログラムの冗長性は高く、理にかなった結果であると言える。

漸化式を1,920画素適用したときの再生シミュレーション像を図4.22に示した。以上より、32 bit float精度で計算した結果と同等の再生像を再生可能なCGH計算が可能な仕様であると言える。



(a) フレネル近似式による再生像。 (b) 回路のシミュレーションによる再生像。

図 4.22: 再生シミュレーション像の比較 (振幅型 CGH).

4.4.2 振幅型の回路構成

振幅型のCGHを計算する専用計算回路のブロック図を図4.23に示した。専用計算回路はPoint Cloud RAM, Recurrence Relation Unit (RRU), RRU controller, OUTPUT FIFOで構成した。

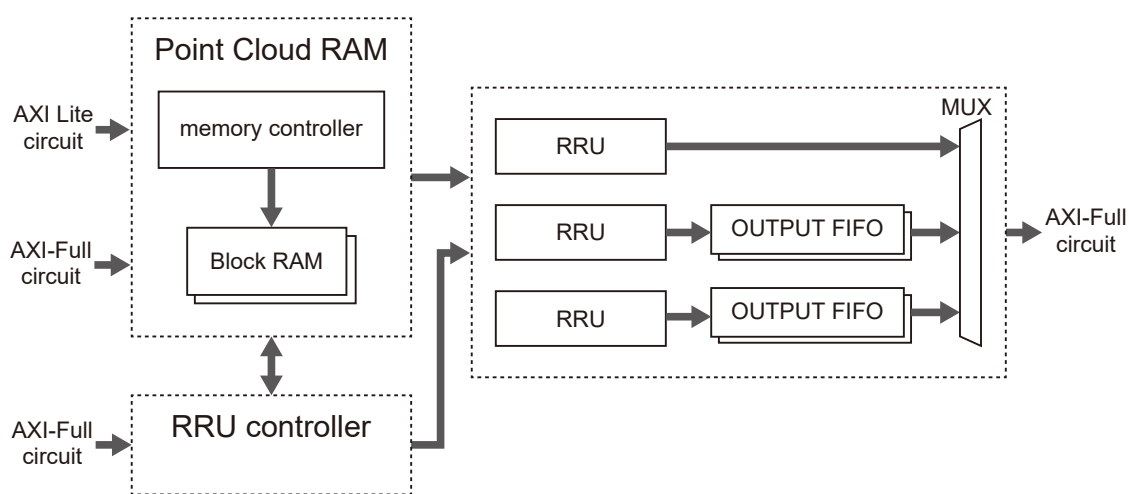


図 4.23: 振幅型電子ホログラフィ専用計算回路のブロック図。

Point Cloud RAMは、点群データをFPGA内部に保存するユニットである。FPGA内部のメモリブロックであるBlock RAMとメモリコントローラで構成した。AXI Fullの回路から入力されるデータの空き状況シグナルをチェックし、空でない場合、データを読み取る。メモリコントローラは入力されたデータをカウントしつつ、Block RAMへと保存する。

RRUは、漸化式法によりCGHの画素値を計算するユニットである。本論文では、3つのRRUが搭載されており、それぞれCGHの横1ライン分である1,920画素を計算する。3つのRRUで5,760画素を一度に計算可能である。RRU controllerは、AXI Liteから入力されるパラメータに応じ、RRUユニットが計算するCGHの座標を計算する。また、Point Cloud Unit内部のBlock RAMから点群データを読み込み、RRUでCGHを計算する。計算結果は、画像出力用のOUTPUT FIFOでバッファリングし、AXI-Full回路へと送る。バッファリングを行うことで、CGHの計算を行うユニット自体はAXIのトランザクションの状態を気にすることなく、AXI-FullのFIFOへ書き込みを行うことができる。書き込まれたデータは、AXI-Fullの回路が計算回路とは非同期で転送を行う。CGH計算回路から見て、AXI-FullのFIFOが一杯になっていないことだけをチェックすればよく、シンプルな回路構成で実装可能である。また、計算時間と転送時間をオーバーラップすることができ、転送時間を隠蔽可能である。

3つのRRUを搭載したが、理想的には3つのRRUのデータをそのまま送ればよい。しかし、そのまま送った場合、メインメモリに書き込まれるCGHデータは整列されておらず、画面表示するにはCPUで整列し直す必要がある。本章のシステムは、そのまま画面表示が可能のように、FPGA内部でCGHの画素を整列する仕様とした。そのため、RRUの計算結果(CGHデータ)をAXI-FullのFIFOへと送る前に、計算結果を一時保存するバッファであるOUTPUT FIFOを設けている。3つのRRUはy座標に対し、並列に計算を実行し、計算結果も同時に出力される。一つめのRRUはそのままAXI-FullのFIFOへと送り、二つめと三つめのRRUからの出力はOUTPUT FIFOへと一時保存する。一つめの転送が終われば、二つめ、三つめとAXI-FullのFIFOへとCGHデータを転送する。こうすることにより、メインメモリに書き込むCGHは整列されたデータとなり、そのまま画面出力が可能である。CPUの介在なしに転送可能なDMA転送の転送速度を生かすことができる。一方で、この仕様の制限により、1つのRRUが一度に計算可能なx軸方向の画素数は、計算したいCGHのx軸方向のサイズと同じである必要がある。

4.4.3 漸化式計算ユニットの実装

漸化式計算ユニットである RRU は、漸化式法により CGH の計算を実行するユニットである。漸化式法を CGH の x 軸方向に関して適応し、 $1,920 \times 1,080$ 画素の CGH 向けに、一度に 1,920 画素の計算を行う。漸化式法における初期位相を計算する Basic Phase Unit (BPU) と加算位相を計算する Additional Phase Unit (APU) から成り立つ。ブロック図を図 4.24 に示した。

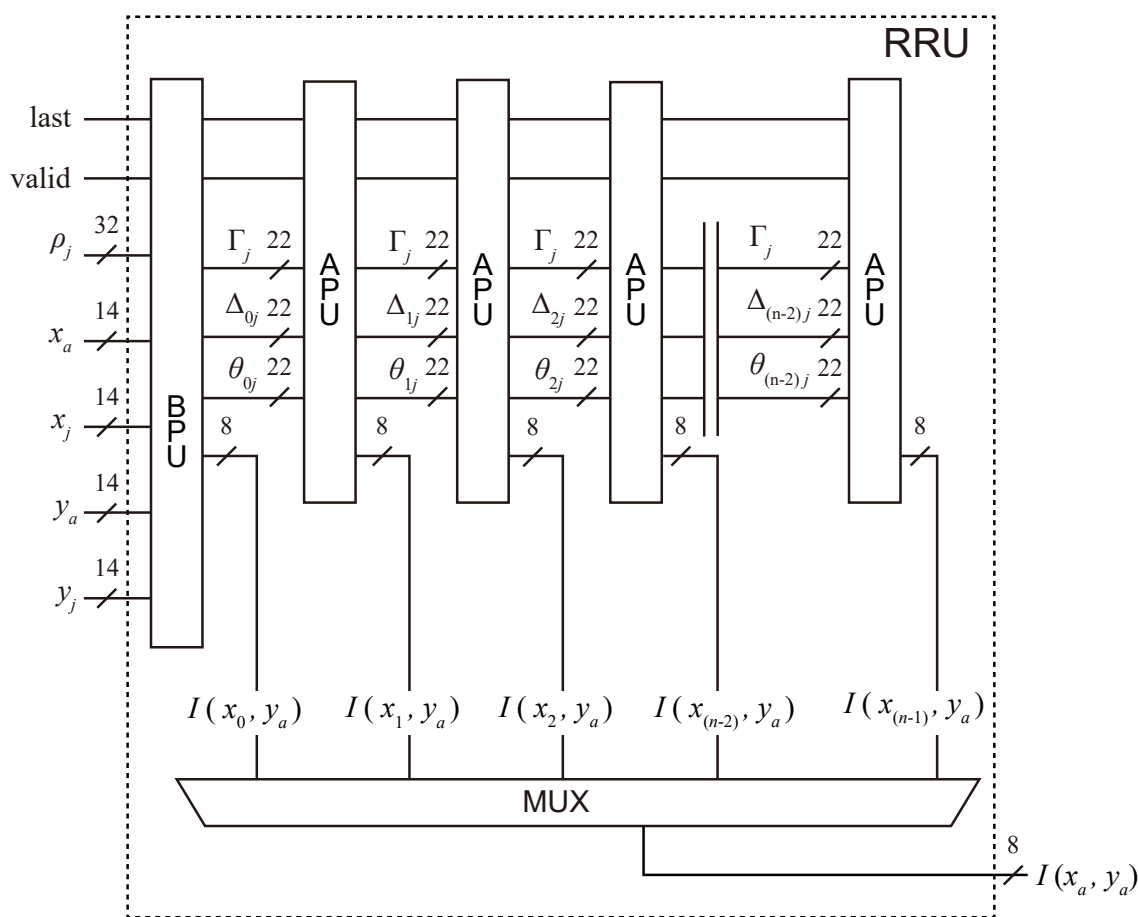


図 4.24: 漸化式計算部のブロック図.

図 4.24 より、RRU は、CGH の座標値である x_a, y_a と点群データである x_j, y_j, ρ_j を入力として CGH 計算を行う。ここで、 ρ_j は、式 (1.17) の $\rho_j = 1/2\lambda|z_j|$ である。

valid と last シグナルは、RRU コントローラによって生成される制御信号である。valid は有効な座標値と点群データが RRU に入力されているとき 1 となる。last は、計算すべき最後の点群データが回路へ入力されたとき 1 となる。これらの制御信号により、最後の点群データを計算したのち、BPU と APU それぞれのユニットは計

算結果を出力する。計算結果の CGH 画素値は、MUX により、ひとつひとつ出力される。

4.4.4 初期位相計算ユニットの実装

初期位相計算ユニットである BPU は、漸化式法における初期位相を計算するユニットである。BPU 内部の計算回路を図 4.25 に示す。BPU は、CGH の座標値である x_a, y_a と点群データである x_j, y_j, ρ_j から、式 (1.17), 式 (1.22), 式 (1.21) の計算を行うために、計算に必要な演算器を並べた。並列に計算可能な演算は並列に演算器を並べ、効率的な計算を可能とした。回路で計算された θ_{aj} は、上位 6 bit が光強度計算ユニット Intensity Unit (IU) へと入力される。IU では、制御信号 valid と last の値により、式 (1.18) に示した θ_{aj} の総和計算を実行し、最終的に CGH 上の 1 画素分の値を出力する。また、次段の APU 向けの θ, Δ, Γ は、それぞれ上位 22 bit を selector で切り出し、出力している。selector 上の [] は、切り出す bit の範囲を表している。

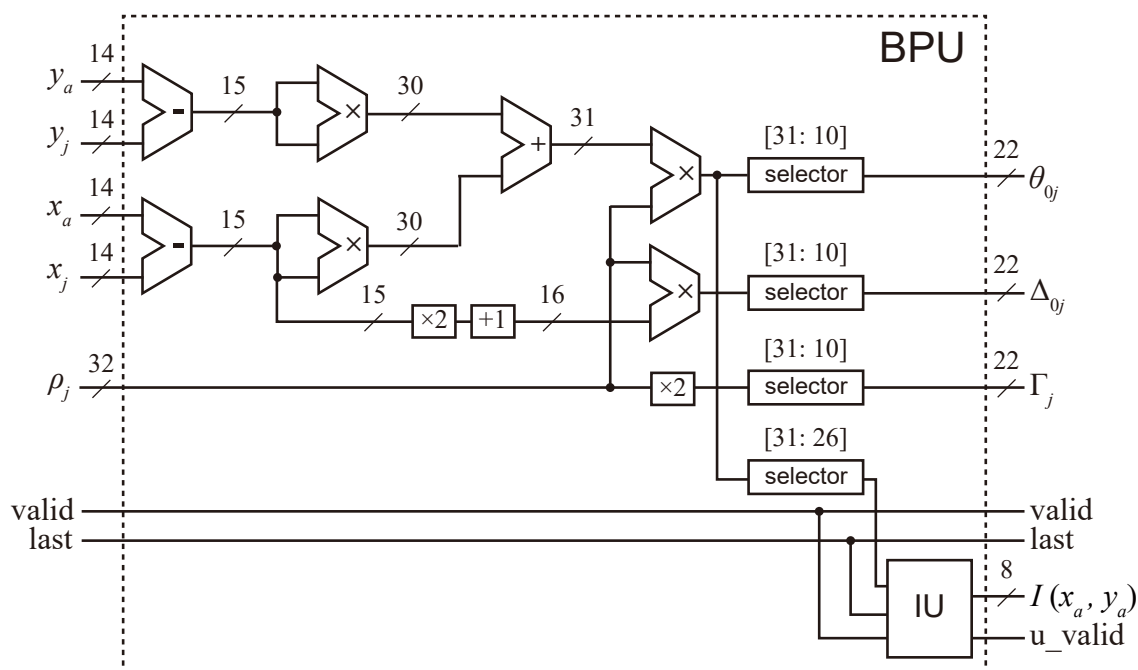


図 4.25: 初期位相計算部のブロック図.

4.4.5 加算位相計算ユニットの実装

加算位相計算ユニットである APU は、漸化式法における加算位相を計算するユニットである。APU 内部の計算回路を図 4.26 に示す。APU は、前段の BPU もしくは APU で計算した $\theta_{(n-2)j}$, $\Delta_{(n-2)j}$, Γ_j を入力として、式 (1.18), 式 (1.24), 式 (1.25) の計算を行う。漸化式法を利用することで式 (1.17) ではなく、式 (1.24), 式 (1.25) の単純な加算のみで求めることが可能である。計算に使用する FPGA リソースを抑えることができ、並列数を増やすことが可能になる。ここで、次段の APU 向けに $\theta_{(n-1)j}$, $\Delta_{(n-1)j}$, Γ_j を出力するが、余弦の周期性から、整数部は無視することができる。したがって、式 (1.24), 式 (1.25) の加算計算による桁上りを見捨てることのできるため、小数部 22 bit だけを出力している。 $\theta_{(n-2)j}$, $\Delta_{(n-2)j}$ の加算による bit 幅増加を気にせず APU を複数段つなげることが可能である。

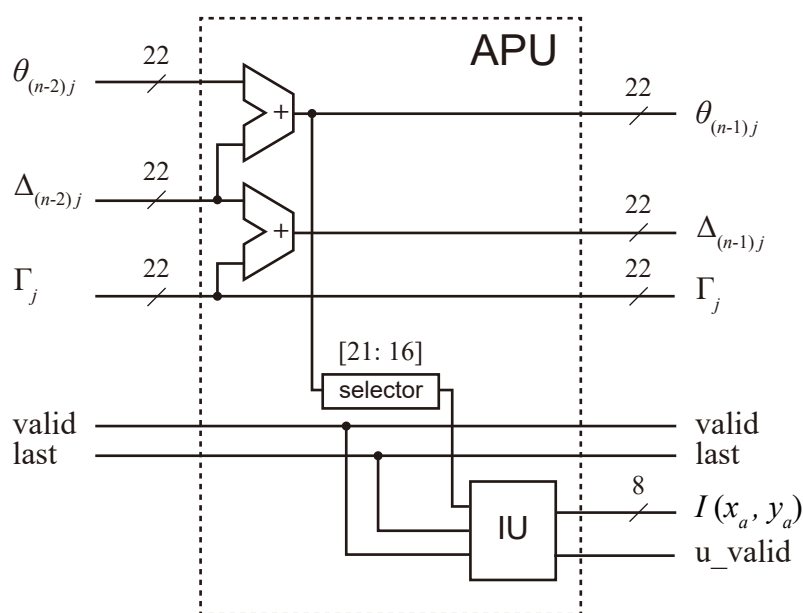


図 4.26: 加算位相計算部のブロック図。

4.4.6 光強度計算ユニットの実装

光強度計算ユニットである IU は、式 (1.18) の計算を行うユニットである。ブロック図を図 4.27 に示した。制御信号 valid と last の値により、 $\theta_{(n-1)j}$ の余弦計算を行い、すべての点群に関して総和計算を実行する。最終的に、総和の正負を表す Most Significant Bit (MSB) により、0 か 255 の画素値を出力する。また、画素値が出力されているとき、u_valid は 1 を出力する。

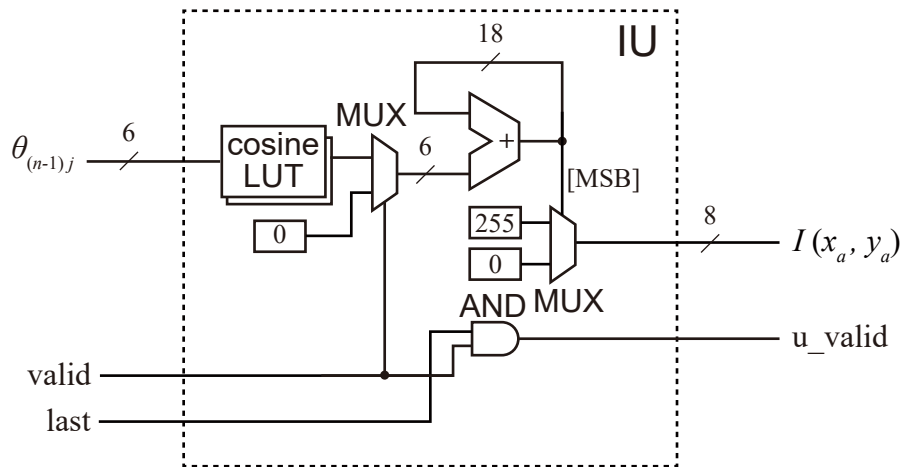


図 4.27: 光強度計算部のブロック図.

4.4.7 リソース使用率

ZCU102を用いて、専用通信回路 (AXI Lite・AXI-Full) と振幅型CGHを計算する専用回路の実装を行った。CGHを計算する回路には3個のBPUと5,757個のAPUを搭載した。1度に5,760画素を計算可能な専用計算機である。計算回路の動作周波数は375 MHzである。FPGAリソースの使用状況を表4.4に示した。

表 4.4: 振幅型CGH専用計算機のリソース使用率.

Resource	Number of usage	Usage[%]
LUT	241,746	88.20
Flip flop	304,587	55.57
Block RAM	3.1 Mb	9.65

4.4.8 計算性能評価

計算時間比較のため、組み込み機器向けGPUであるNVIDIA Jetson TX1との比較を行った。実測した結果を図4.28に示す。式(1.18)のフレネル近似式と漸化式法を利用したCGH計算プログラムをCUDA 10.2を用いて実装した。また、参考値として、デスクトップ向けCPUであるIntel Core i9-9900Kとの比較も行った。CPU環境は、Ubuntu 18.04.4 LTS (Linux Kernel 5.3.0-46-generic) が動作するシステムで、Intel C compiler 19.0.5.281を用いた。フレネル近似式と漸化式法をfloat精度

と int 精度でそれぞれ実装し比較した。また、CGH の高速化アルゴリズムとして代表的な N-LUT [75] との比較も行った。CPU, GPU とともにすべての計算コアを利用し、並列に実行した。

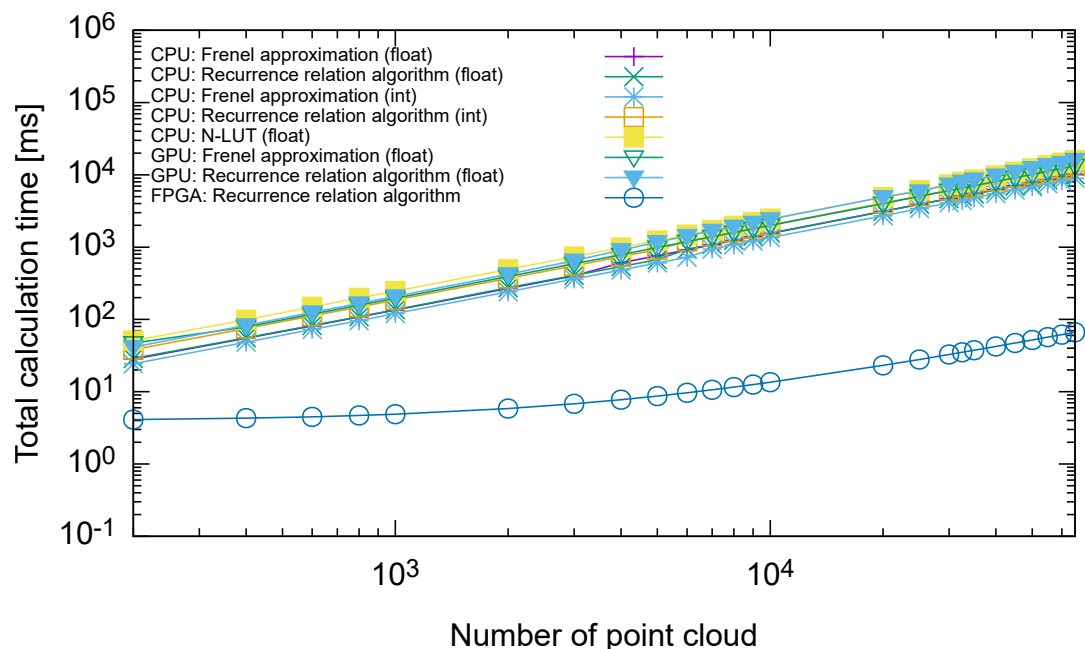


図 4.28: 計算時間の比較.

図 4.28 より、CPU では int 精度で実装したフレネル近似式が最も高速であった。組み込み機器向け GPU では、フレネル近似式での実装が高速であった。点群数が 50,000 点での組み込み機器向け GPU と CPU それぞれ最速の計算時間と ZCU102 との比較を表 4.5 に示した。

表 4.5: 振幅型 CGH 計算時間の比較.

Hardware	Calculation time [ms]	fps
FPGA (ZCU102)	52	19
GPU	10,135	0.10
CPU	6,863	0.15

図 4.28 において、CPU や GPU と比べ ZCU102 は線形なグラフになっていないことがわかる。これは、専用計算機の実装によるものである。本章で開発した専用計算機において、RRU は完全に並列に動作する。一方で、RRU 内部の BPU と APU はパイプライン実行するため、完全に並列に動作せず、パイプラインディレイと呼

ばれる遅延時間がある。本章の実装では、計算する点群数がBPUとAPUの総数よりも小さい場合は、並列計算コアを十分に活用できず、性能が出ない仕様となっている。計算する点群数がBPUとAPUの総数よりも大きい場合（ここでは数万点の点群を入力した場合）、並列動作とみなすことができる。

ZCU102におけるCGH 1フレームあたりの理論的な計算時間は、パイプラインディレイを無視した場合、式(4.1)で与えられる。

$$t_{zcu102} [s] = M \times K \div 5,760 \text{ 並列} \div 375 \text{ MHz} \quad (4.1)$$

ここで M は点群数、 K はホログラムの画素数を示す。点群数を5万点、ホログラムサイズを200万画素とすると、0.05秒程度であり、ほぼ理論値通りの結果が得られていることがわかる。表4.5より、組み込み機器向けのGPUと比べ190倍の高速化を達成した[42]。また、デスクトップ向けのCPUと比較しても131倍の高速化を達成した。

4.4.9 光学再生

開発した計算機を用いて点光源データからCGHを計算し、光学再生を行った。SLMは振幅変調型で、画素ピッチ6.4 μm 、1,920 \times 1,080画素である。再生照明光には532nmの緑色レーザーを用いた。再生像はSLMから0.5mの位置に再生するように設定し、デジタルカメラで撮影した。284点の点光源で表されたキューブ(図4.29)、11,646点の点光源で表された恐竜の骨格(図4.31)、44,647点の点光源で表されたチェス盤(図4.33)を縦軸に関して少しずつ回転させながらCGHを計算し、実際に光学再生を行った動画のスナップショットを図4.30 - 図4.34に示した。

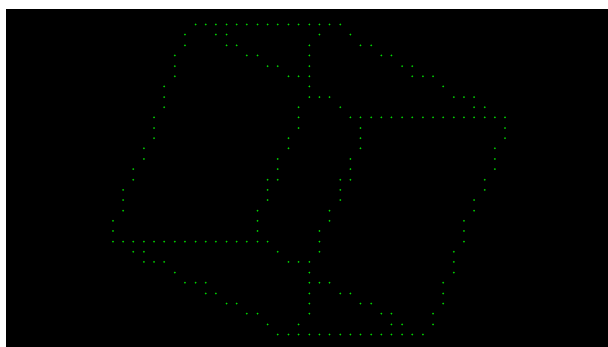


図 4.29: 点光源データの様子 (キューブ).

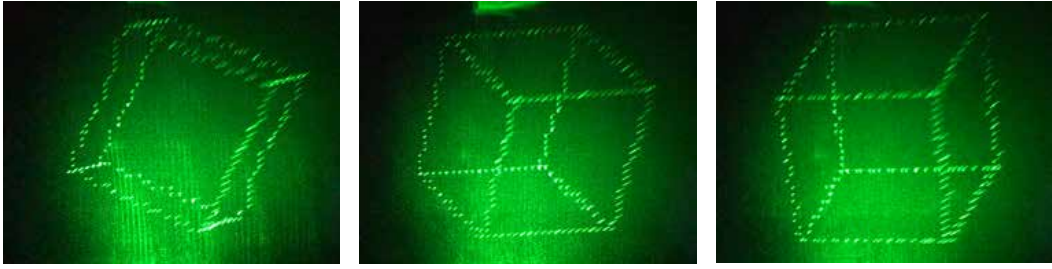


図 4.30: 光学再生した動画の一部 (キューブ).

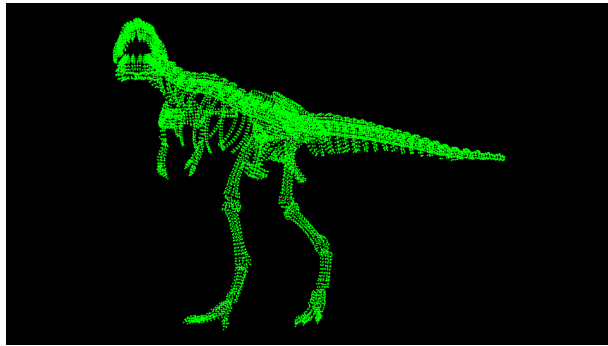


図 4.31: 点光源データの様子 (恐竜の骨格).



図 4.32: 光学再生した動画の一部 (恐竜の骨格).

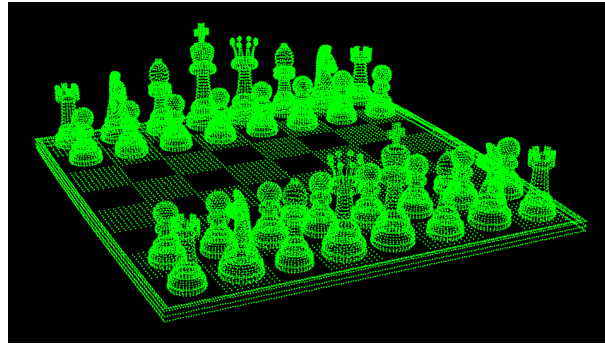


図 4.33: 点光源データの様子 (チェス盤).



図 4.34: 光学再生した動画の一部 (チェス盤).

4.5 位相型電子ホログラフィ専用計算回路

漸化式法を利用した位相型 CGH 計算回路を FPGA に実装した. 先行研究 [36] では 1 チップ 320 並列 (漸化式の適用画素数が 320 画素間隔) の位相型 CGH を計算する専用計算機の開発がされた. 本研究では, 振幅型の専用計算機と同様, 位相型の専用計算機でも計算精度の再検証と大規模 FPGA でさらなる並列計算を可能とするアーキテクチャの設計を行った.

4.5.1 位相型の計算精度設定

位相型でも 4.4.1 と基本は同様の精度を利用した. x_a, x_j, y_a, y_j は符号あり 14 bit 入力, ρ_j は符号無し 32 bit 入力とした. 漸化式法における $\theta_j, \Delta_{aj}, \Gamma_{aj}$ は符号あり 22 bit とした.

ここで, 位相型 CGH 計算では式 (1.19) より, 余弦と正弦計算が必要であるが, 振幅型同様 LUT 法を利用した. 位相型 CGH の計算に必要な余弦と正弦の LUT 精度は先行研究で調査されており [36], ここでも 6 bit 入力, 6 bit 出力の LUT を利用した.

式 (1.19) の逆正接計算にも同様に LUT を利用した。ただし、必要なテーブルサイズは、点群数が増えるにしたがって取り得る範囲が大きくなるため、容量の大きな LUT テーブルを作成する必要がある。先行研究では、複素振幅分布の実部と虚部をアドレスとして入力する際に、実部と虚部の絶対値が大きい方を基準として正規化することで必要なテーブルサイズを削減した [36]。実部と虚部 5 bit ずつの 10 bit を入力し、8 bit 出力の LUT が利用されていた。

図 4.35 に正規化の例を示す。符号付きの数値は、2 進数表記では 2 の補数表現で表す。このとき、最上位ビットを符号をあらわすビットと考えることができる。実部および虚部の絶対値を比較するためには、それぞれのビットについて、上位ビットから順番に符号ビットと比較し、符号ビットと同じビットがどの桁まで続いているかを調べる。符号ビットと同じビットが連続する数が少ないほど、絶対値の大きな値となる。図 4.35 では、実部は 10 bit 分、虚部は 8 bit 分符号ビットと同じビットが続いているため、虚部の方が絶対値が大きいことがわかる。絶対値が大きい方を基準として、符号ビットと同じビットを最上位とし、任意のビット数をとり出し、逆正接計算用の LUT 入力に利用する。このとき、もう一方の数値からも同じ位置からビットを抽出する。ここで取り出した値が正規化された値としてテーブルのアドレスとなる。このようにして正規化を行うことによって、必要な LUT テーブルサイズを削減可能である。本研究でも実部と虚部 5 bit ずつの 10 bit を入力し、8 bit 出力の LUT を利用した。

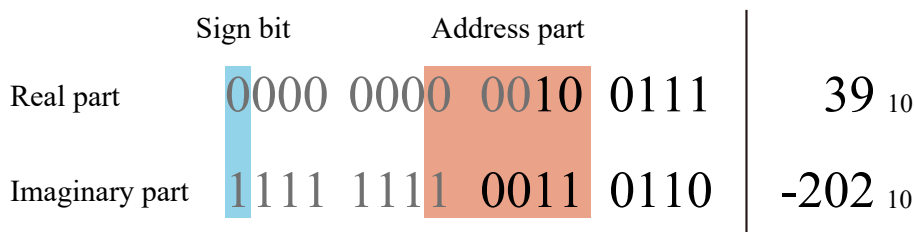


図 4.35: 正規化の例.

CGH 計算の精度に問題が無いことをシミュレーションベースで検証した結果を示す。32 bit float 精度で計算した CGH と比較した CGH 同士の平均 2 乗誤差 (MSE) と、CGH から再生計算した再生像を画像評価指標の一種である PSNR を指標として比較した結果を図 4.36 に示した。SSIM との比較を図 4.37 に示した。65,000 点の点群から CGH を計算し、再生シミュレーション条件は、画素ピッチ 6.4 μm, 1,920 × 1,080 画素, 再生照明光波 532 nm, 再生距離 0.5 m である。

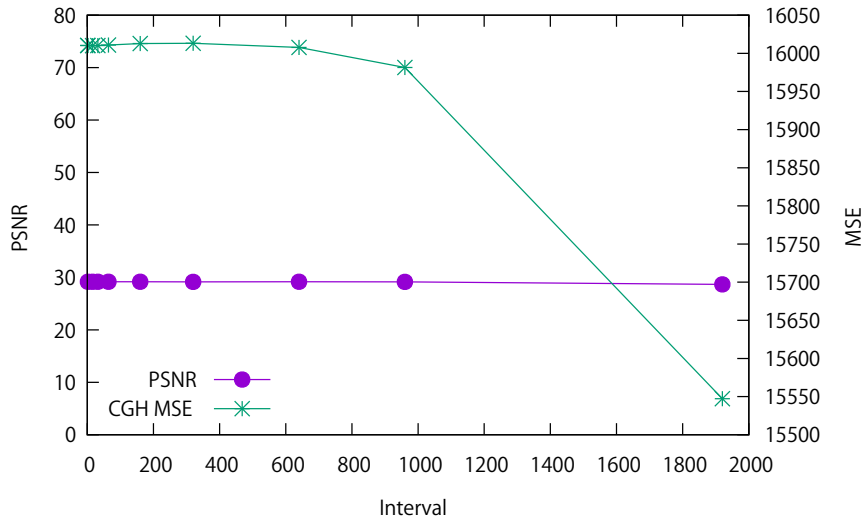


図 4.36: 再生像の PSNR と CGH の MSE 比較 (位相型).

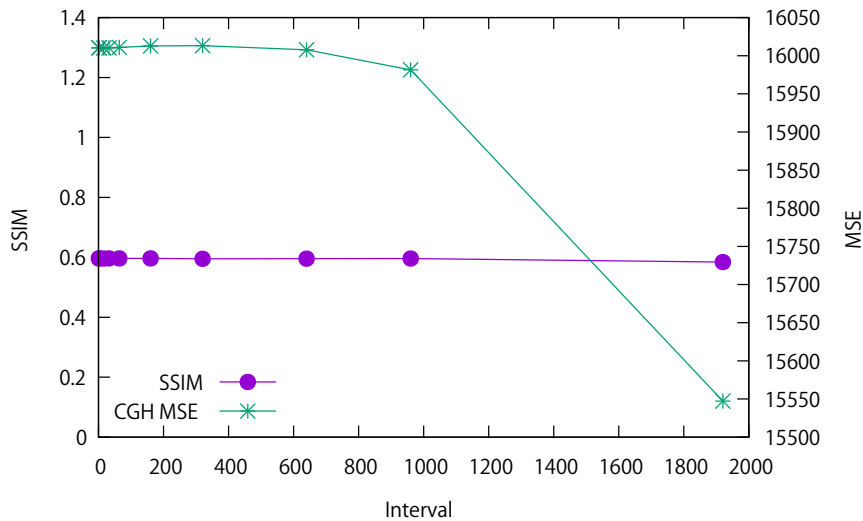
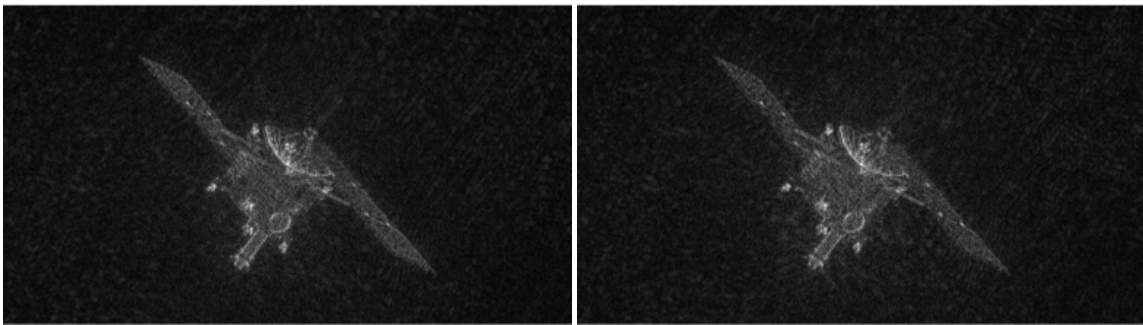


図 4.37: 再生像の SSIM と CGH の MSE 比較 (位相型).

図 4.36, 図 4.37 では, 漸化式の適用間隔と各値の変化を示した. 振幅型と比べ, 漸化式の適用間隔の変化に合わせて, 一見 CGH 自体の誤差が減っているように見えるが, MSE の変化は 600 程度であり, どちらかというともあまり変わっていないと考えることができる. 再生像の画質も劣化しておらず, 1,920 画素適用時, PSNR は 37.3 dB, SSIM は 0.846 と, 横すべての画素に関して漸化式法を適用しても問題ないことがわかる.

漸化式を 1,920 画素適用したときの再生像を図 4.38 に示した. 以上より, 32 bit float 精度で計算した結果と同等の再生像を再生可能な CGH 計算が可能な仕様であ

ると言える。



(a) フレネル近似式による再生像.

(b) 回路のシミュレーションによる再生像.

図 4.38: 再生シミュレーション像の比較 (位相型 CGH).

4.5.2 位相型の回路構成

漸化式法を利用した CGH 計算回路を FPGA に実装した. 位相型の CGH を計算する専用計算回路のブロック図を図 4.39 に示した. 専用計算回路は, 振幅型同様, Point Cloud RAM, Recurrence Relation Unit (RRU), RRU controller で構成した. それぞれの役割は, RRU は位相型 CGH の計算を行うことを除いて, 振幅型と同様である. RRU は, 漸化式法により CGH の画素値を計算するが, 計算に必要なリソースが増加しており, 1つの RRU のみ搭載した. 1つの RRU によって, 1,920 画素を一度に計算可能である.

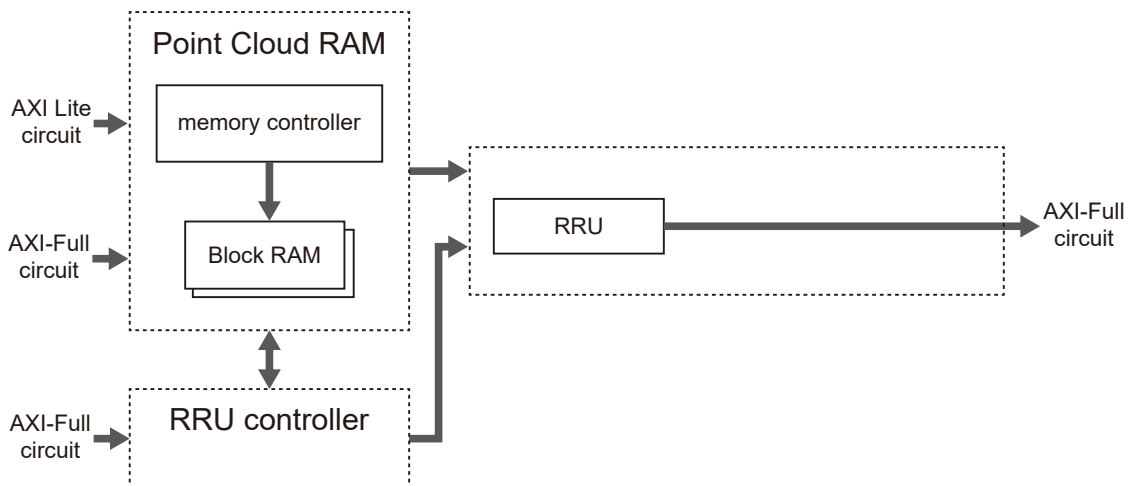


図 4.39: 位相型電子ホログラフィ専用計算回路のブロック図.

3つのRRUを搭載した振幅型の回路では、CGHデータの整列問題によって、1つのRRUが一度に計算可能なx軸横方向の画素数は、計算したいCGHのx軸横方向のサイズと同じである必要があった。1つの場合はその問題は無いため、ある程度自由な並列数を選択可能である。ただし、AXI-FullのFIFOへデータを送る関係上、RRUの並列数は計算したいCGHの横幅と同じか約数である必要がある。本章では、 $1,920 \times 1,080$ 画素のCGH向けに、1,920並列とした。1,920は4K Ultra HDディスプレイで利用されている横幅3,840画素の約数なため、そのまま4Kディスプレイ向けのCGHを計算することが可能である。

4.5.3 漸化式計算ユニットの実装

漸化式計算ユニットであるRRUは、漸化式法によりCGHの計算を実行するユニットである。漸化式法をCGHのx軸方向に関して適応し、 $1,920 \times 1,080$ 画素のCGH向けに、一度に1,920画素の計算を行う。漸化式法における初期位相を計算するBPU、加算位相を計算するAPU、複素振幅分布の正規化を行うNormalization Unit (NU)、偏角計算用LUTであるarctan LUTを搭載した。ブロック図を図4.40に示した。

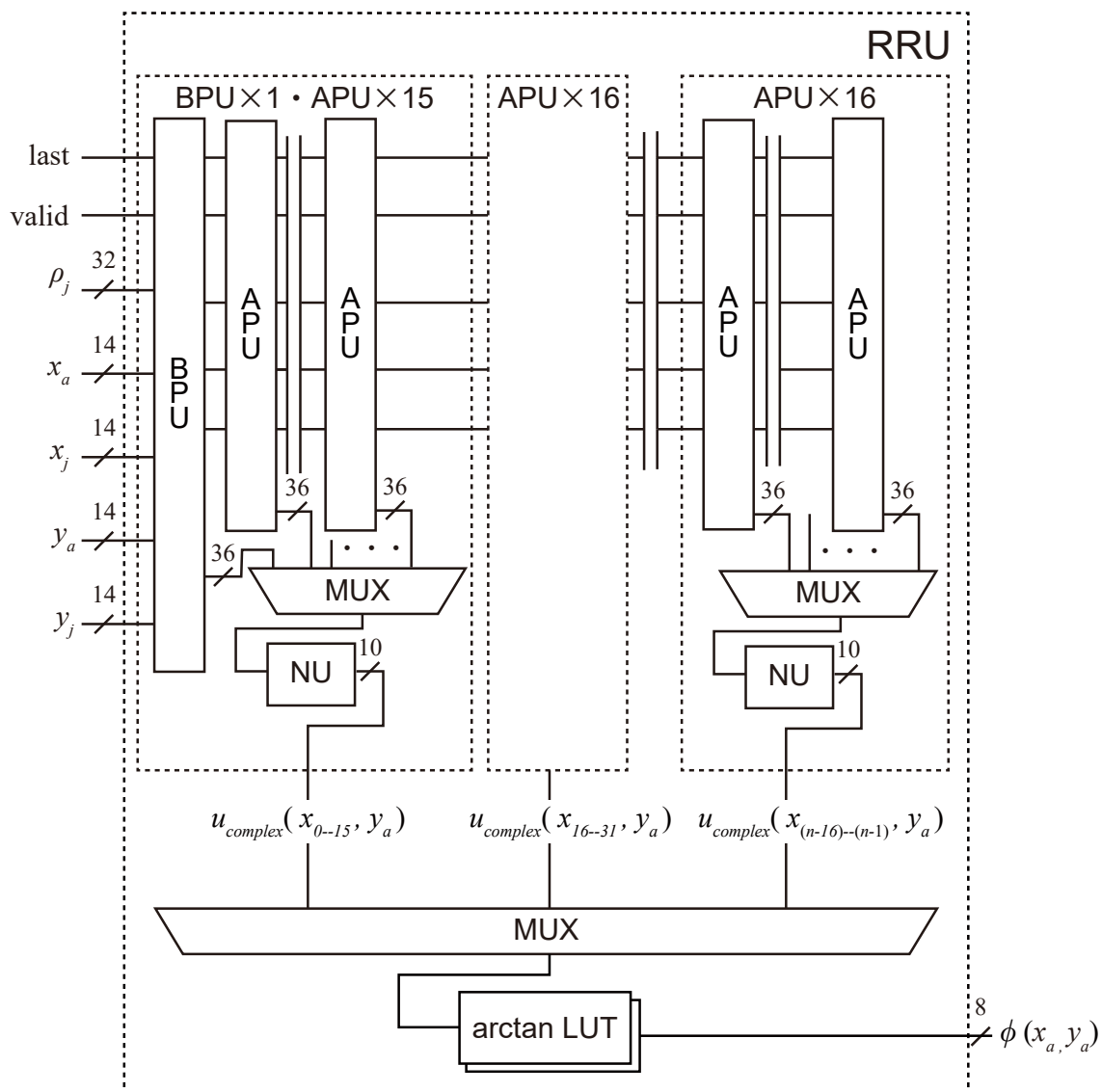


図 4.40: 漸化式計算部のブロック図.

RRU は、CGH の座標値である x_a, y_a と点群データである x_j, y_j, ρ_j を入力として複素振幅分布の計算を行う。振幅型同様、 $\rho_j = 1/2\lambda |z_j|$ は組み込み CPU を用いて計算を行い、リソースの増加や動作周波数の低下を抑えた。

ここで、BPU と APU に対して NU と arctan LUT をどのような比率で搭載するかという問題がある。シンプルな構成では、BPU と APU に対して NU と arctan LUT を 1 つずつ搭載する。この場合、RRU の MUX は振幅型同様 8 bit × 1,920 の切り替えを行えばよい。一方で、arctan LUT を 1,920 個も搭載するには FPGA 内部のメモリが足りない。NU と arctan LUT の数を最も小さくする構成では、RRU1 つに対し、NU と arctan LUT をそれぞれ 1 つ搭載する構成である。しかし、MUX は 36(実部 + 虚部) bit × 1,920 の切り替えが必要になり、タイミング制約が厳しくな

る。また、信号伝達用の配線を多く消費する。このようにBPUとAPUに対してNUと \arctan LUTをどのような比率で搭載するかは難しい問題である。本研究では、必要なメモリサイズ、FPGAのタイミング制約を考慮し、16並列ごとにNUを1つ搭載し、 \arctan LUTはRRUに対し1つだけ搭載した。FPGA自体のアーキテクチャによって最適な比率は変わり、将来的に別のFPGAへ移植する際は、構成の再検討を行う必要がある。

4.5.4 初期位相計算ユニットの実装

初期位相計算ユニットであるBPUは、漸化式法における初期位相を計算するユニットである。BPU内部の計算回路を図4.41に示す。BPUは、CGHの座標値である x_a, y_a と点群データである x_j, y_j, ρ_j から、式(1.17)、式(1.22)、式(1.21)の計算を行うために、計算に必要な演算器を並べた。並列に計算可能な演算は並列に演算器を並べ、効率的な計算を可能とした。回路で計算された θ_{0j} は、上位6bitが複素振幅計算ユニット Complex amplitude Unit (CAU) へと入力される。CAUでは、制御信号 valid と last の値により、式(1.18)に示した θ_{aj} の総和計算を実行し、複素振幅分布 u_{complex} を出力する。

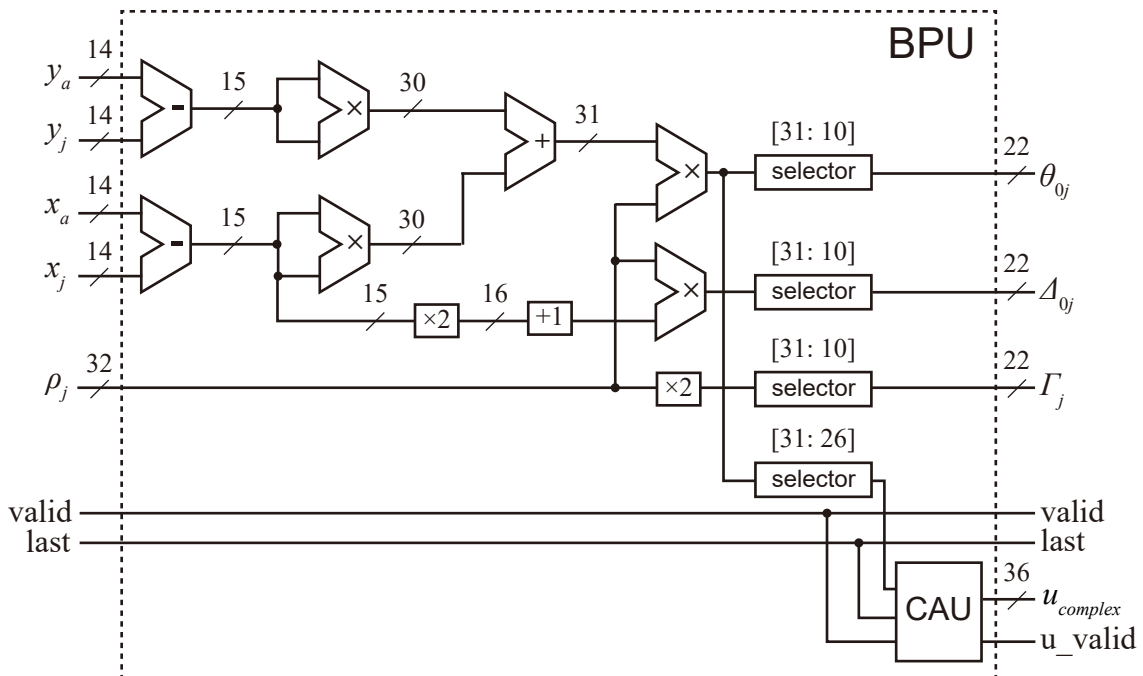


図 4.41: 初期位相計算部のブロック図.

4.5.5 加算位相計算ユニットの実装

加算位相計算ユニットである APU は、漸化式法における加算位相を計算するユニットである。APU 内部の計算回路を図 4.42 に示す。APU は、前段の BPU もしくは APU で計算した $\theta_{(n-2)j}$, $\Delta_{(n-2)j}$, Γ_j を入力として、式 (1.18), 式 (1.24), 式 (1.25) の計算を行う。漸化式法を利用することで式 (1.17) ではなく、式 (1.24), 式 (1.25) の単純な加算のみで求めることが可能である。計算に利用する FPGA リソースを抑えることができ、並列数を増やすことが可能になる。次段の APU 向けに $\theta_{(n-1)j}$, $\Delta_{(n-1)j}$, Γ_j を出力するが、余弦と正弦の周期性から、整数部は無視することができる。したがって、式 (1.24), 式 (1.25) の加算計算による桁上りを見捨てることのできるため、小数部 22 bit だけを出力している。 $\theta_{(n-2)j}$, $\Delta_{(n-2)j}$ の加算による bit 幅増加を気にせず APU を複数段つなげることが可能である。

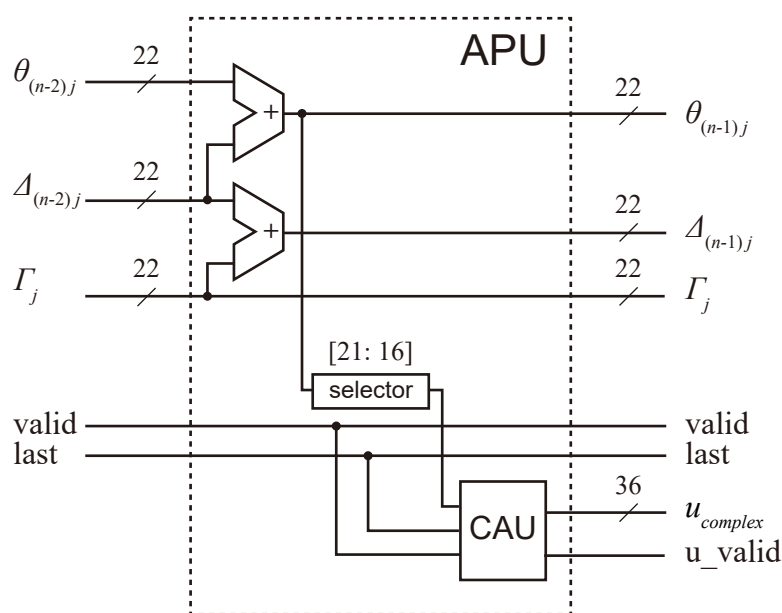


図 4.42: 加算位相計算部のブロック図。

4.5.6 複素振幅計算ユニットの実装

複素振幅計算ユニットである CAU は、式 (1.19) 内の \cos と \sin の総和計算を行うユニットである。ブロック図を図 4.43 に示した。制御信号 valid と last の値により、 $\theta_{(n-2)j}$ の余弦・正弦計算を行い、すべての点群に関して総和計算を実行する。CAU は総和計算後、 u_valid から 1 を出力し、実部 (Re) と虚部 (Im) それぞれ 18 bit ずつ計算結果の出力を行う。

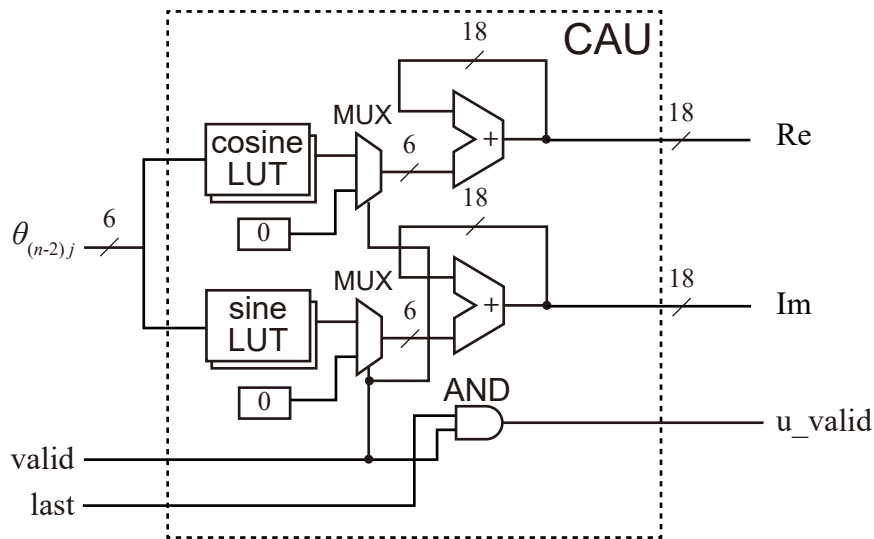


図 4.43: 複素振幅分布計算部のブロック図.

4.5.7 リソース使用率

ZCU102 を用いて、専用通信回路 (AXI Lite・AXI-Full) と位相型 CGH を計算する専用回路の実装を行った。CGH を計算する回路には 1 個の BPU と 1,919 個の APU を搭載した。1 度に 1,920 画素を計算可能な専用計算機である。計算回路の動作周波数は 375 MHz である。FPGA リソースの使用状況を表 4.6 に示した。

表 4.6: 位相型 CGH 専用計算機のリソース使用率.

Resource	Number of usage	Usage[%]
LUT	177,604	64.80
Flip flop	249,030	45.43
Block RAM	2.89 Mb	8.99

4.5.8 計算性能評価

計算時間比較のため、組み込み機器向け GPU である NVIDIA Jetson TX1 との比較を行った。実測した結果を図 4.44 に示す。式 (1.19) のフレネル近似式と漸化式法を利用した CGH 計算プログラムを CUDA 10.2 を用いて実装した。また、参考値として、デスクトップ向け CPU である Intel Core i9-9900K との比較も行った。CPU 環境は、Ubuntu 18.04.4 LTS (Linux Kernel 5.3.0-46-generic) が動作するシステム

で、Intel C compiler 19.0.5.281 を用いた。フレネル近似式と漸化式法を float 精度と int 精度でそれぞれ実装し比較した。また、CGH の高速化アルゴリズムとして代表的な N-LUT [75] との比較も行った。CPU、GPU とともにすべての計算コアを利用し、並列に実行した。

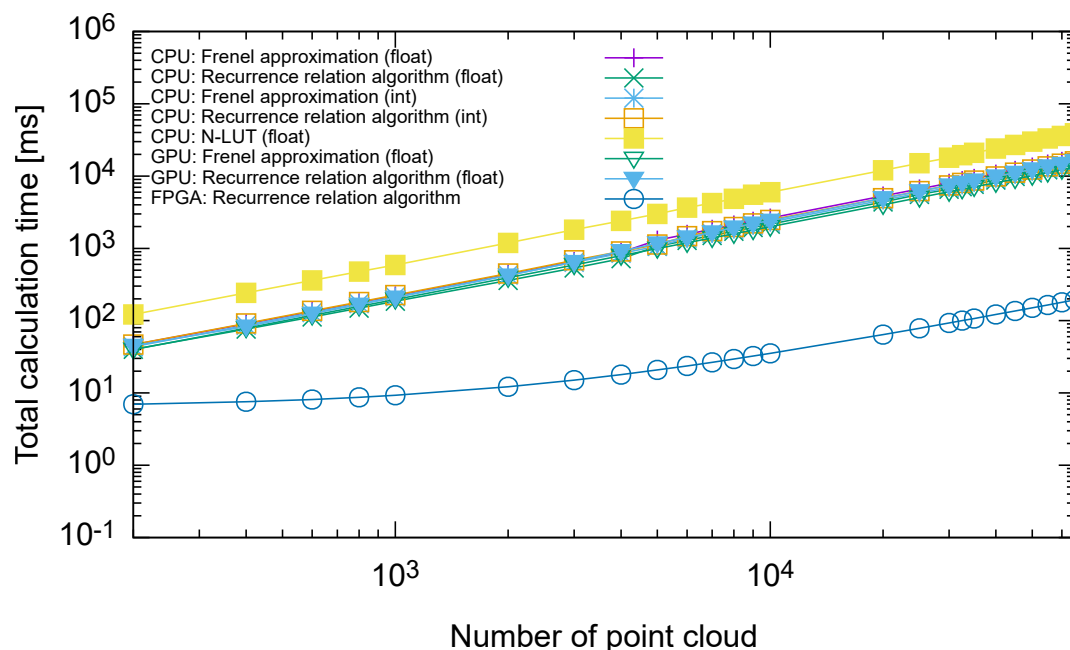


図 4.44: 計算時間の比較.

図 4.44 より、CPU では float 精度で実装した漸化式法が最も高速であった。組み込み機器向け GPU では、フレネル近似式での実装が高速であった。点群数が 50,000 点での組み込み機器向け GPU と CPU それぞれ最速の計算時間と ZCU102 との比較を表 4.7 に示した。

表 4.7: 位相型 CGH 計算時間の比較.

Hardware	Calculation time [ms]	fps
FPGA (ZCU102)	150	6.67
GPU	10,139	0.10
CPU	11,014	0.09

図 4.44 において、CPU や GPU と比べ ZCU102 は線形なグラフになっていないことがわかる。これは、振幅型同様、パイプラインディレイによるものである。ZCU102 における位相型 CGH 1 フレームあたりの理論的な計算時間は、パイプラインディレイ

イを無視した場合，式 (4.2) で与えられる．

$$t_{zcu102} [s] = M \times K \div 1,920 \text{ 並列} \div 375 \text{ MHz} \quad (4.2)$$

ここで M は点群数， K はホログラムの画素数を示す．点群数を 3 万点，ホログラムサイズを 200 万画素とすると，0.15 秒程度であり，ほぼ理論値通りの結果が得られていることがわかる．表 4.7 より，組み込み機器向けの GPU と比べ 67 倍の高速化を達成した [76]．また，デスクトップ向けの CPU と比較しても 73 倍の高速化を達成した．

4.5.9 光学再生

開発した計算機を用いて点光源データから CGH を計算し，光学再生を行った．SLM は位相変調型で，画素ピッチ $8.4 \mu\text{m}$ ， $1,920 \times 1,080$ 画素である．再生照明光には 532 nm の緑色レーザーを用いた．再生像は SLM から 0.5 m の位置に再生するように設定し，デジタルカメラで撮影した．284 点の点光源で表されたキューブ (図 4.45)，11,646 点の点光源で表された恐竜の骨格 (図 4.47)，44,647 点の点光源で表されたチェス盤 (図 4.49) を縦軸に関して少しずつ回転させながら CGH を計算し，実際に光学再生を行った動画のスナップショットを図 4.46 - 図 4.50 に示した．

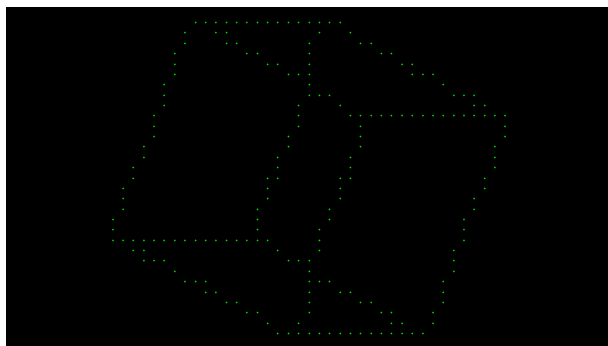


図 4.45: 点光源データの様子 (キューブ)．



図 4.46: 光学再生した動画の一部 (キューブ)．

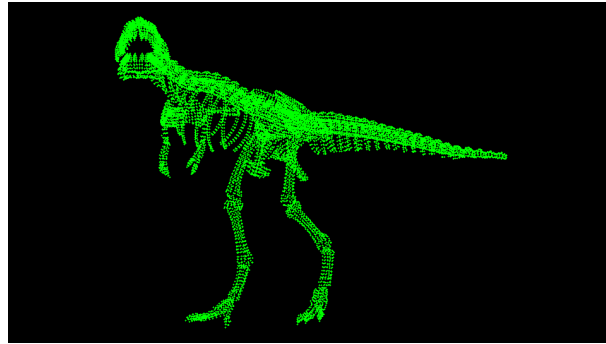


図 4.47: 点光源データの様子 (恐竜の骨格).



図 4.48: 光学再生した動画の一部 (恐竜の骨格).

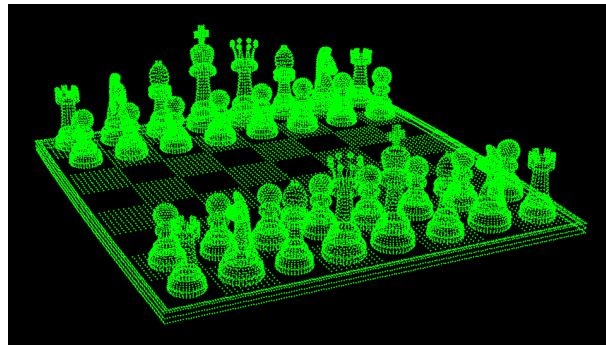


図 4.49: 点光源データの様子 (チェス盤).



図 4.50: 光学再生した動画の一部 (チェス盤).

4.6 電子ホログラフィを用いたインタラクティブシステム

先行研究ではGPUを用いてインタラクティブシステムの開発が行われた [77]. 本章では, Leap Motion と ZCU102 を用いて開発した専用計算機システムを利用し, インタラクティブシステムの開発を行った. 開発したインタラクティブシステムでは, 手の動きに合わせて3次元像をインタラクティブに操作可能である.

4.6.1 Leap Motion について

Leap Motion は, Leap Motion 社が提供する手の3次元トラッキングが可能な入力機器である (図 4.51). 2基の赤外線カメラと赤外線照射LEDが内蔵されており, 両手と10本の指をそれぞれ独立して同時にトラッキング可能である. Leap Motion を利用することで, 手の動きをコンピュータ内で再現し, 手の動きに合わせて, 3D 物体を動かすといったインタラクティブな操作が可能となる.



図 4.51: Leap Motion の外観.

4.6.2 Leap Motion を用いたインタラクティブシステム

インタラクティブシステム全体の構成を 図 4.52 に示した. システムの構成は, Leap Motion を利用したトラッキング部, 光学系部, ZCU102 を利用した計算部に分けられる.

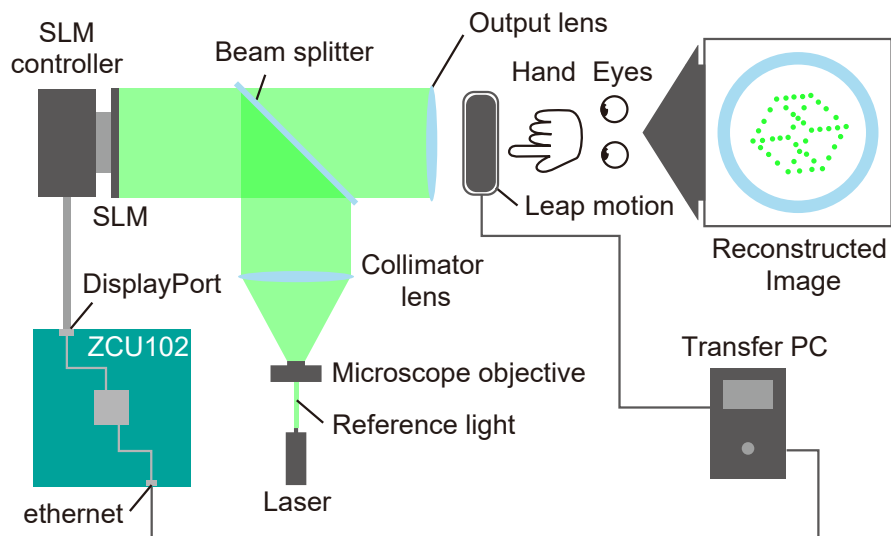


図 4.52: 電子ホログラフィ専用計算機を利用したインタラクティブシステムの構成.

トラッキング部では、Leap Motion を使用し、人差し指先の位置を取得している。Leap Motion は、直接 ZCU102 接続することも可能であるが、実験時は Leap Motion 社から提供されている Linux 用ドライバが古く動作しなかったため、Windows が動作する PC を別に用意して接続を行った。Leap Motion は、双方向通信を低コストで実行可能な Web Socket サーバとして動作可能である。ZCU102 の組み込み CPU 上で Web Socket クライアントを動作させ、Web Socket により Windows PC に接続した Leap Motion から指の位置情報を取得し、インタラクティブに動かすためのパラメータとした。

光学系部では、シンプルな電子ホログラフィ再生用の光学系を利用した。SLM には画素ピッチ $6.3 \mu\text{m}$ 、解像度 $1,920 \times 1,080$ 画素の振幅変調型 LCD を利用した。光源は、 532 nm のレーザーを利用した。指の動きと 3 次元再生像を撮影するために、Output Lens を観察部分に挿入した。カメラでの撮影は困難になるが、Output Lens 無しでも肉眼で 3 次元像は観察可能である。

計算部である ZCU102 内部の構成に関して 図 4.53 に図示した。組み込み CPU は、FPGA の制御、CGH を計算するための点群データの管理などを行い、FPGA に構築した専用計算回路を計算アクセラレータとして利用することで高速に処理可能である。

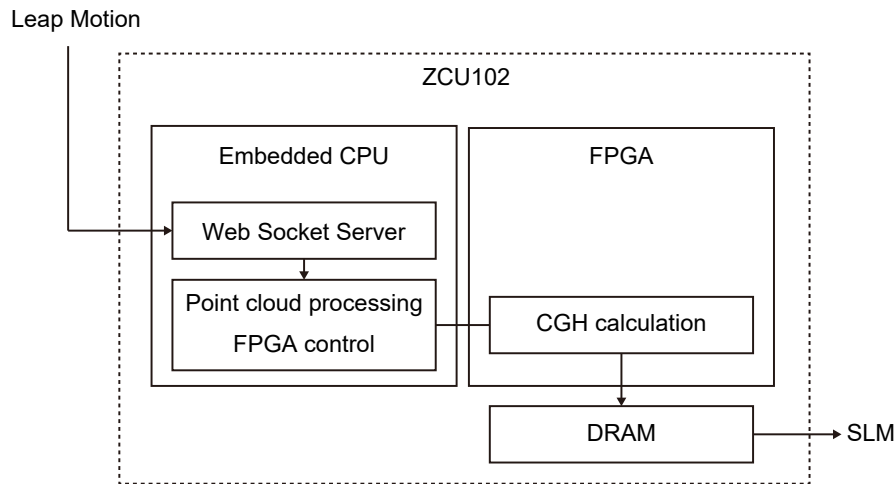


図 4.53: インタラクティブ処理システム (専用計算機システム内).

組み込み CPU では、CGH をインタラクティブに変化させるためのデータ取得と専用計算回路の制御を実行した。データ取得のため、Web Socket クライアントが動作し、ネットワーク越しに Leap Motion から座標データを取得する。取得した座標値を元にして、移動後の点群座標を計算し、FPGA に構築した専用計算機へ点群データの転送を行う。

FPGA に構築した専用計算回路は、3,840 並列 (BPU×2, APU×1919×2) の回路を利用した。動作周波数は 375 MHz である。計算結果である CGH の画素値は、物理メモリ上のフレームバッファへ FPGA から直接書き込みを行った。このフレームバッファに書き込まれたデータは、ZCU102 に搭載されている映像端子へ出力が行われる。FPGA からフレームバッファへ直接書き込みを行うことで、画像データ転送のための遅延は発生しない構成とした。

Leap Motion 用の Linux ドライバの制限のため、別に用意した PC に Leap Motion を接続して利用した。Web Socket を利用し、取得したデータを処理し、表示する部分までは、スタンドアロンなシステムの開発を行った。インタラクティブに再生像を操作している様子を 図 4.54 に示した。

4.6.3 インタラクティブシステムの処理時間

Leap Motion を用いたインタラクティブシステムにおいて、システム全体の性能評価のため、65,000 点の点群から $1,920 \times 1,080$ 画素の CGH を単位時間あたり何フレーム処理可能かを検証した。その結果、60 s あたり 600 フレームの CGH を処理できていた。すなわち、10 fps で CGH が処理可能なシステムであると言える。

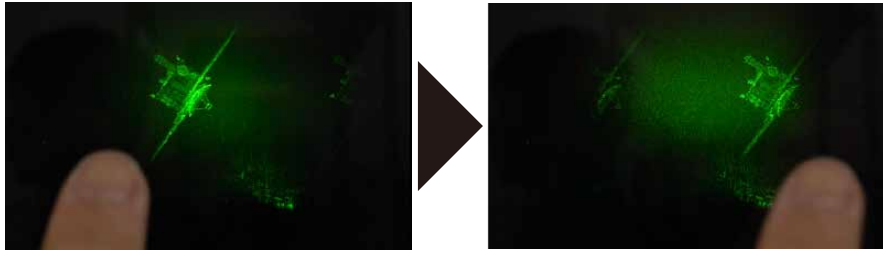


図 4.54: インタラクティブに3次元映像を操作している様子. 指の動きに合わせて再生像が移動している.

10 fps という結果は, FPGA のみでの計算時間とほぼ等しい. これは組み込み CPU と専用計算回路の協調動作により, 点群処理にかかる時間が上手く遮蔽された結果である. また, CGH の表示も CPU を介さず直接出力しているため, データ転送にかかる遅延が存在しない高速なシステムであるといえる.

4.7 小括

本章では, 振幅型と位相型の CGH を計算する電子ホログラフィ専用計算機の開発を行った. 特に, HMD での電子ホログラフィ応用を視野にいれた組み込み CPU と FPGA がワンチップに搭載された SoC を用いて, 小さなシステムの開発を行った. 研究の初期段階として, Xilinx 社が提供する Zynq UltraScale+ MPSoC ZCU102 評価キットを利用した. 組み込み CPU では独自にカスタマイズした OS を動かし, 点群処理などを行わせた. ワンチップ上の FPGA には CGH 専用計算回路を構築し, 組み込み CPU から計算アクセラレータとして利用することで, SoC レベルでの高速化を実現した.

FPGA に構築した回路には, AXI プロトコルに対応する通信回路と漸化式法を利用した CGH 計算回路を実装した. 振幅型専用回路は, 375 MHz で動作する 5,760 の専用計算コアにより, 50,000 点の点群から $1,920 \times 1,080$ 画素の CGH を 19 fps で計算可能である. 位相型専用回路は, 375 MHz で動作する 1,920 の専用計算コアにより, 50,000 点の点群から $1,920 \times 1,080$ 画素の CGH を 6.67 fps で計算可能である.

組み込みシステム向けの SoC としては, NVIDIA 社が GPU を搭載した SoC である Jetson シリーズを提供している. 本章のシステムは Jetson TX1 と比べ, 振幅型専用計算機は 131 倍, 位相型専用計算機は 67 倍の高速化を達成した. 参考値として, デスクトップ向けの GPU である NVIDIA RTX 2080 Ti では同様の条件で, 振幅型 CGH 計算に 220 ms, 位相型 CGH には 230 ms 程度かかる. 本章では, SoC での比較を行ったが, デスクトップ向けの GPU と比較しても本章の SoC システムは, 振幅型で 4 倍, 位相型で 1.5 倍の高速化を実現した.

また、SoC全体の性能を評価するため、Leap Motion社が提供する手の3次元トラッカーであるLeap Motionを利用し、インタラクティブシステムを開発した。本章のインタラクティブシステムは、再生像を指で動かすといった簡単なシステムではあるが、将来的にHMDに組み込み、再生像(3D映像)を動かすといったアプリケーションで有効に動作することを示した。

本章の目的であるHMD向けに本システムを利用することを考えたとき、重要なのはフレームレートである。必要なフレームレートは表示する映像によって大きく変化する。映画は24 fpsであり、地上デジタル放送では29.97 fpsである。ゲーム用途では144 fps、240 fpsのディスプレイが市販されている。一方で、最低限動画として認識されるためには数fpsでも十分であると考えられる。従って、本章で開発したシステムはSoCでの実装ながら、 $1,920 \times 1,080$ 画素のSLMで上限近い点光源数5万点を動画として表示可能な性能を達成したと言える。

今後の展望として、本章で利用したSoCと表示用のディスプレイが1つの基板に搭載された専用ボードの開発を行っていききたい。計算機とディスプレイが一体化することで、組み込み機器としての利便性は高まる。実際にHMDへ組み込みを行うなどして、電子ホログラフィを利用したHMDの評価を行っていききたい。

第5章 Alveo U250を用いた電子ホログラフィ専用計算機

本章の目的は、電子ホログラフィ計算の高速化を達成することである。先行研究では、電子ホログラフィ専用計算機 HOlographic ReconstructioN (HORN) [35, 36, 38–40, 67–69] をはじめとした専用計算機の開発が行われ、電子ホログラフィ計算における有効性が示された。これまでに発表された HORN は、独自に開発を行った専用基板を利用した専用計算機である。本章の専用計算機は、Xilinx 社が提供する Alveo U250 データセンターアクセラレータカードを利用し、専用計算機システムの構築を行った。Xilinx 社が提供する製品としての FPGA ボードを利用した専用計算機を開発することにより、電子ホログラフィ実用への指針を示した。

5.1 電子ホログラフィ専用計算機の構成

本章では、電子ホログラフィ専用計算機開発に、Xilinx 社が提供する Alveo U250 データセンターアクセラレータカード (以下、U250 と呼ぶ) を用いた。U250 の外観を図 5.1、搭載チップ内の FPGA 部分に関する仕様を表 5.1 に示した。U250 は、データセンター向けのアクセラレータであり、表 5.1 に示したように、膨大な FPGA リソースを持っている。また、U250 では、FPGA 内部のメモリブロックである Block RAM に加え、UltraRAM が利用可能である。UltraRAM は Block RAM と比べ、データ読み込みに必要なレイテンシが増加しているが、容量は 4 倍近く多い。本章の専用計算機では、UltraRAM を点群や CGH の保存に利用し、Block RAM は arctan LUT など LUT に利用した。



図 5.1: Alveo U250 データセンターアクセラレータカードの外観. 外装に覆われており, FPGA チップは確認できない.

表 5.1: U250 の論理回路リソース.

Resource	Available
Look-up Table	1,728,000
Flip Flop	3,456,000
Block RAM	94.5 Mb
UltraRAM	360 Mb
DSP	12,288

5.1.1 専用計算システムの構成

専用計算機システム開発のため, PC・FPGA ボード間の通信回路開発と専用計算回路の開発を行った. U250 はデータセンター向けのアクセラレータボードであり, 図 5.2 に示すように, PC の PCI Express ポートに拡張ボードとして接続して使用することが想定されている.

基本構成は, ZynqMP で行った専用計算機開発と同じく, PC (CPU) から FPGA に構築した専用回路を計算アクセラレータとして利用する構成とした. CGH の計算では, 除算を含む $\rho_j = 1/2\lambda |z_j|$ の計算は CPU を用いて計算を行い, リソース使用量の削減, 動作周波数の向上をおこなった.

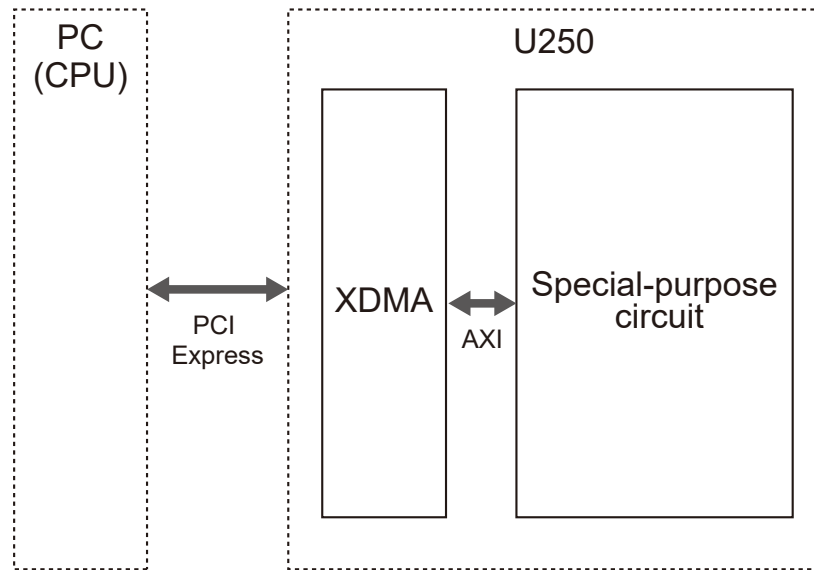


図 5.2: 専用計算システム全体の構成.

5.2 PC・FPGA ボード間通信

PC・FPGA ボード間は、PCI Express ポートを通じて通信を行う。U250 では、Xilinx 社よりリリースされている統合ソフトウェアプラットフォームである Xilinx Vitis を利用可能である。C、C++ 記述から通信回路や所望の回路を自動生成する高位合成による開発が可能である。一方で、ZynqMP のときと同じく、数千～数万の計算コアが生成する計算結果を効率良く転送することで、高速な専用計算機システムを有効活用することが可能である。本章では、通信回路最適化のため、通信回路の開発も行った。

PCI Express 用の通信回路自体は、Xilinx 社から DMA for PCI Express (PCIe) Subsystem (XDMA) として IP コアが提供されている。XDMA を利用することで、PCI Express の通信は意識する必要はないが、FPGA 内部の回路とは AXI プロトコルで通信する仕様のため、AXI に対応した通信回路を実装する必要がある。また、ZynqMP のときと異なり、XDMA の場合、XDMA 側に AXI 通信の制御権 (controller) があり、開発する計算回路は responder として実装する必要がある。

5.2.1 通信回路の実装

PC・FPGA 間通信は、XDMA を通して AXI プロトコルに変換される。本章では、すべての通信を AXI-Full によって行う実装とした。AXI-Full では、高速な通信のた

め、データを連続して送受信するバースト転送とアドレスとデータの送受信を並行して実行するオーバーラップ転送が実行可能である。本章の通信回路も、バースト転送とオーバーラップ転送が可能な仕様とした。開発した回路のブロック図を図5.3に示した。

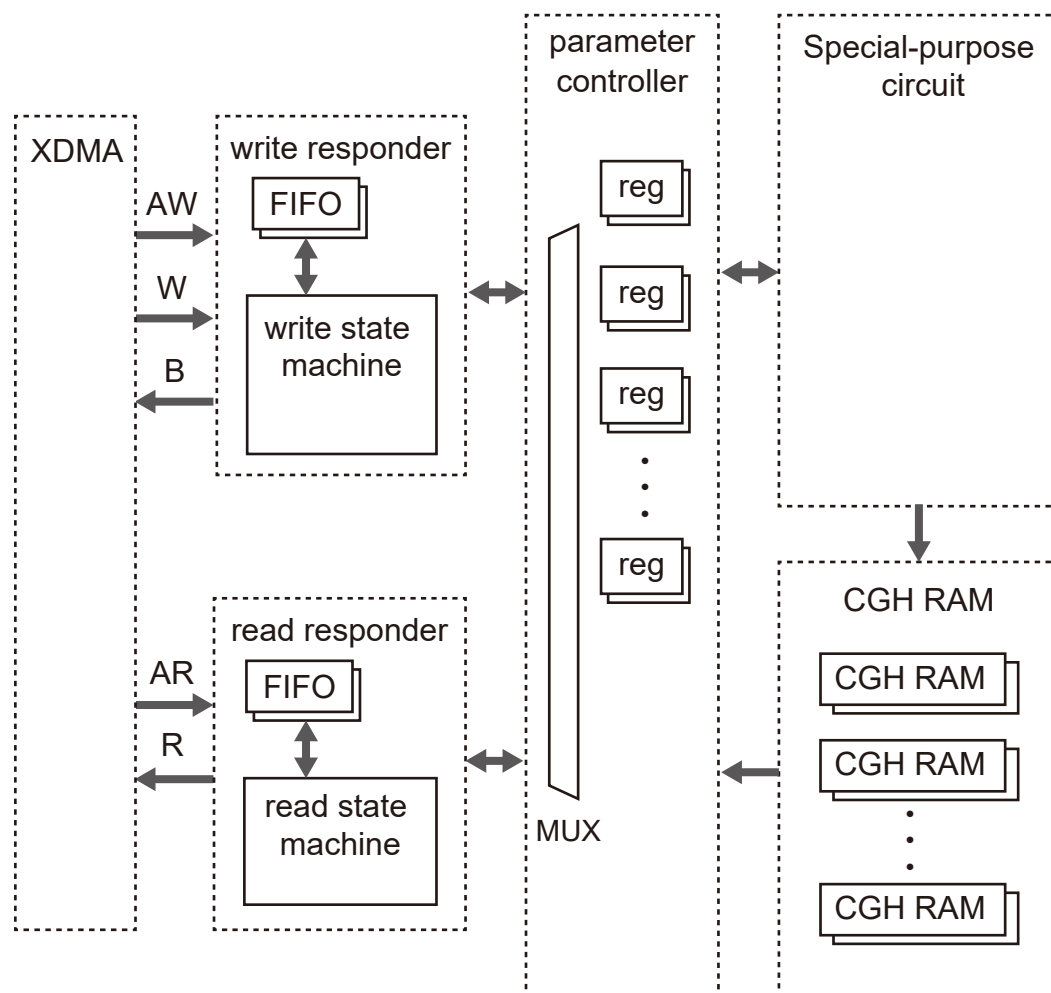


図 5.3: AXI 通信回路の様子。

開発した AXI 通信回路は、XDMA からの書き込み処理に応答する write responder と読み取り処理に応答する read responder, 各 responder で取得したアドレスから、データの割り振りと読み込みを行う parameter controller, 計算結果の CGH を一時保存する CGH RAM で構成した。XDMA からの書き込み応答のため、write responder には、AW Channel と W Channel と B Channel を接続し、読み込み応答のため、read responder には、AR Channel と R Channel を接続した。write responder と read responder は、オーバーラップ転送対応のため、アドレスとバースト長を一時保存す

る FIFO とステートマシンで構成した。write responder の書き込み応答の場合は、アドレスとデータを取得し、parameter controller へと送信する。parameter controller は、アドレスによって、制御信号か点群データかを判定し、レジスタにそのまま書き込むか、専用計算回路側へとデータをさらに送信する。ここで、レジスタは専用計算回路制御用のレジスタであり、専用計算回路の制御信号へと接続してある。XDMA の設定では AXI のデータ幅は最大 512 bit である。本章では通信回路の安定を優先し、32 bit で開発を行った。

5.2.2 write responder の実装

WRITE ステートマシンは、オーバーラップ転送とバースト転送対応のため、アドレスのやり取りを行うステートマシンと、データのやり取りを行うステートマシンで構成した。アドレス用のステートマシンを図 5.4 に、データ用のステートマシンを図 5.5 に示した。

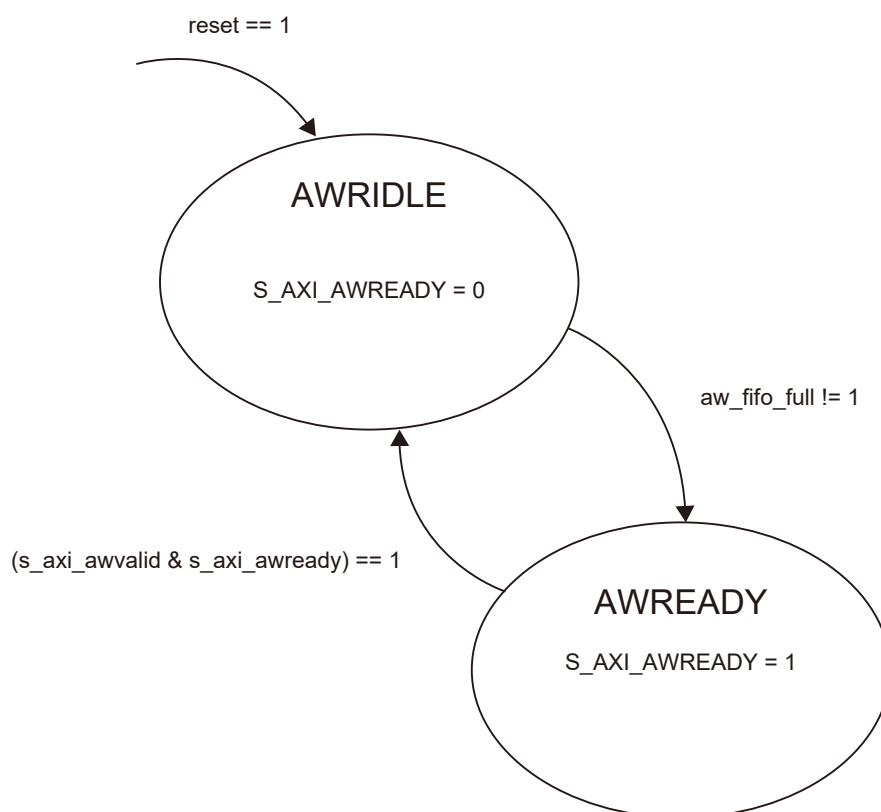


図 5.4: write responder (アドレス転送用) の遷移図.

図 5.4 の状態マシンでは、PC から転送されてきたアドレスとバースト長を FIFO に保存するために、2つの状態を実装した。アドレス保存用の FIFO が空の場合、AWREADY へ遷移し、アドレスデータが受信可能であることを示す S_AXI_AWREADY を 1 にする。S_AXI_AWVALID が 1 になったとき、受け取ったアドレスデータを FIFO へ保存し、S_AXI_AWREADY を 0 に戻し、AWRIDLE へ戻る。

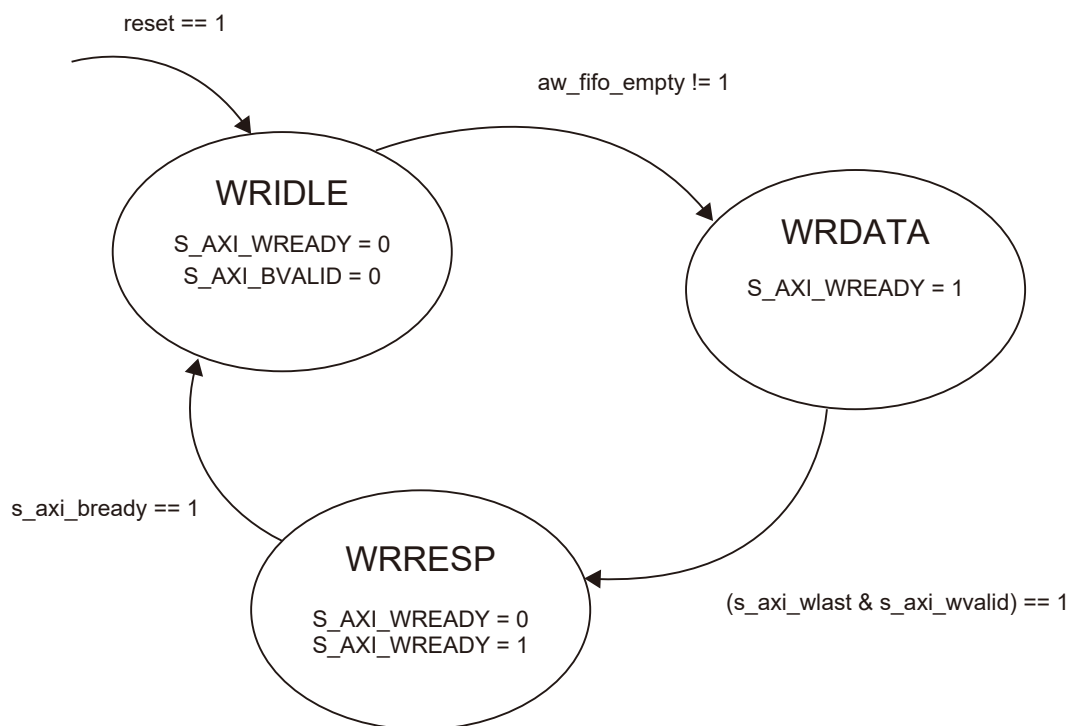


図 5.5: write responder (データ転送用) の遷移図.

図 5.5 の状態マシンでは、PC から転送されてきたデータを FPGA 内部へ保存するために、3つの状態を実装した。図 5.3 中のアドレスデータ保存用 FIFO にデータがある場合 ($aw_fifo_empty \neq 1$)、状態は WRDATA へ遷移する。WRDATA では、 s_axi_wlast と s_axi_wvalid がともに 1 になるまで、すなわちバースト転送一回分のデータをすべてを受信したとき、WRRESP へ遷移する。WRRESP は、受信応答として s_axi_bready を 1 にして WRIDLE へ戻る。ここで、受信応答は常に成功を返すシンプルな実装としている。

5.2.3 read responder の実装

READ ステートマシンは、オーバーラップ転送とバースト転送対応のため、WRITE ステートマシンと同様、アドレスのやり取りを行うステートマシンと、データのやり取りを行うステートマシンで構成した。アドレス用のステートマシンを図 5.6 に、データ用のステートマシンを図 5.7 に示した。

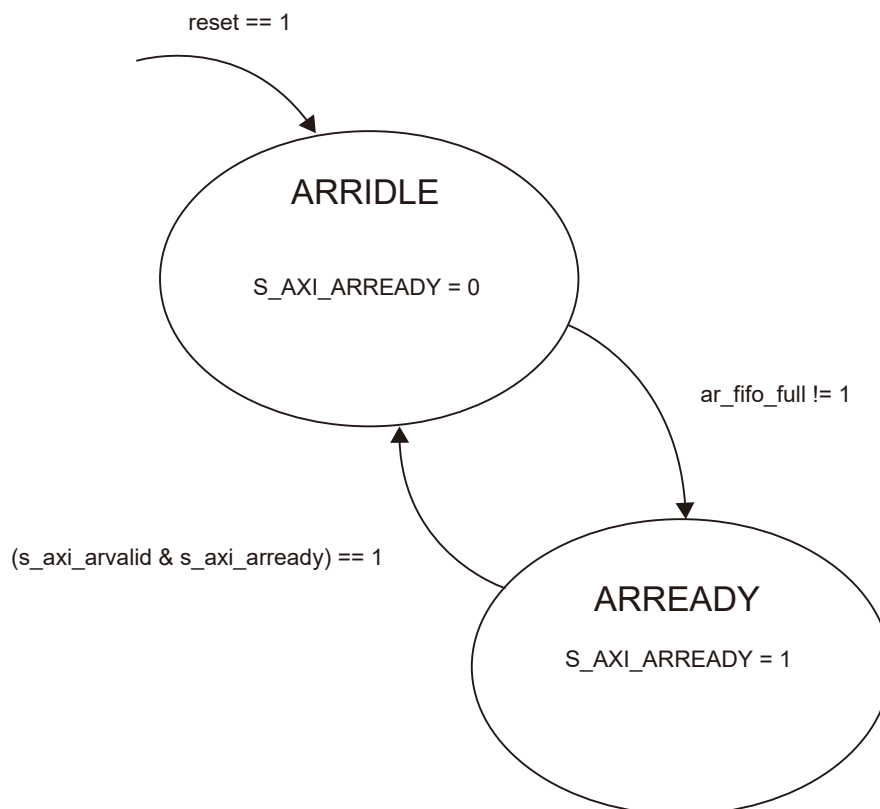


図 5.6: read responder (アドレス転送用) の遷移図.

図 5.6 のステートマシンでは、PC から転送されてきたアドレスとバースト長を FIFO に保存するために、図 5.4 同様、2つのステートを実装した。アドレスデータの転送は write, read の処理に関わらず、PC から FPGA 側へ転送されるため、動作も図 5.4 と同じである。

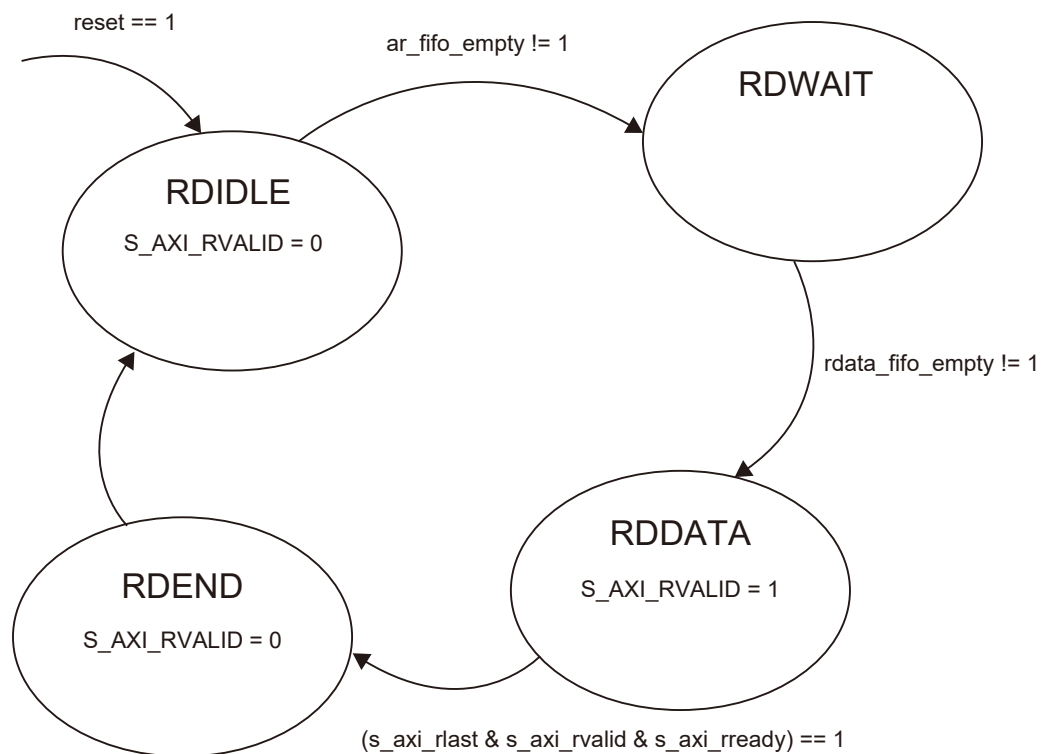


図 5.7: read responder (データ転送用) の遷移図.

図 5.7 のステートマシンでは、PC から転送されてきたデータを FPGA 内部へ保存するために、4 つのステートを実装した。write responder と異なり、アドレス保存用の FIFO に加え、転送データバッファリング用 FIFO がある。まず、アドレスデータ保存用 FIFO にデータがある場合 ($ar_fifo_empty \neq 1$)、ステートは RDWAIT へ遷移する。RDWAIT では、アドレス保存用の FIFO からとり出すアドレスデータに対応したデータをバッファリング用の FIFO へ一時保存する。AXI のトランザクションでは、有効なデータを出力していることを示す valid 信号が 1 を出力し続けている場合でも、受信可能であることを示す ready 信号が 1 になったのちに 0 になる場合がある。ready 信号が途中で変化しても正しく転送可能なように、FIFO にデータを一時保存することで対応した。データ保存用 FIFO にデータがあることを確認したのち、ステートは RDDATA へ遷移する。RDDATA では、データ保存用 FIFO からデータを取り出し、FPGA 内部から PC へデータ転送を実行する。一回分のバースト転送が完了したのち、ステートは RDEND へ遷移し、そのまま RDIDLE へ戻る。

5.2.4 アドレスマップ

CPU (PC) から見た FPGA 内部のアドレスとレジスタの名前について表 5.2 に示した。Local signal の [] は、ビットの位置もしくは範囲を示す。Local signal に ”_reg” がついたものが書き込み可能なレジスタである。他は読み取り専用である。reserved は利用できないことを示す。

clc_ready は計算回路がアイドル状態であることを示す。clc_start_reg は、計算開始をコントロールするレジスタである。x_max_reg は、計算する CGH の横幅を指定する。y_max_reg は、計算する CGH の高さを指定する。pcl_num_reg は、計算する点群数を指定する。y_loop_reg は、1 つの RRU が Y 軸方向に繰り返す計算回数を指定する。例として、RRU が 8 つ搭載されている場合、 $1,920 \times 1080$ 画素の CGH 計算を行う場合、135 を指定する。RRU の繰り返し回数を CPU から指定可能にすることで、RRU の数が増え変わった場合でも柔軟な計算が可能な仕様とした。点群データは、4096 から $x_0, y_0, z_0, a_0, x_1, y_1, z_1, a_1, \dots, x_j, y_j, z_j, a_j$ の順で書き込みを行う。 a_j は点光源の振幅である。本研究では 1 に固定しているが、将来的に振幅値を指定可能なようにアドレス空間を用意した。

計算結果である CGH の画素値はアドレス 4,096 番地から読み取ることで得ることができる。ここで、RRU の数だけ 4,194,304 ずつ離れたアドレス空間が生成される。CPU からデータを読み取る場合は、まず 4,096 番地から 259,200 アドレス読み取り (4 画素 \times 64,800 回読み取り)、4,198,400 番地までアドレスを進め、また 259,200 アドレス読み取るという動作を RRU の搭載数分繰り返す。ここで、点群用のアドレス空間も含め、アドレスが飛び飛びになっているのは、AXI のバースト転送の仕様によるものである。AXI でバースト転送をする際、4 KB の境界を跨いではいけない。この対処法は、ハードウェアで対処する方法、ドライバで対処する方法などがあるが、本研究ではシンプルな実装のため、特に対策せず、アドレスマップを 4 KB の境界に並べることで 4 KB 境界を跨ぐようなバースト転送を行わない実装とした。

表 5.2: アドレスマップ (U250).

Address	Register Name	Local signal
0	CLC_FLAG	[1]:clc_ready,[0]:clc_start_reg
4	NONE	reserved
8	X_MAX	[31:0]:x_max_reg
12	NONE	reserved
16	Y_MAX	[31:0]:y_max_reg
20	PCL_LEN	[31:0]:pcl_num_reg
24 32	NONE	reserved
36	Y_LOOP_NUM	[31:0]:y_loop_reg
40 4,092	NONE	reserved
4,096 1,048,576	PCL_RAM_BASE	[31:0]a_reg / [31:0]z_reg / [31:0]y_reg / [31:0]x_reg
4,096 4,198,396	CGH0	[31:24]cgh,[23:16]cgh,[15:8]cgh,[7:0]cgh
4,198,400 8,392,700	CGH1	[31:24]cgh,[23:16]cgh,[15:8]cgh,[7:0]cgh
8,392,704 : :	CGH2	[31:24]cgh,[23:16]cgh,[15:8]cgh,[7:0]cgh

5.3 振幅型電子ホログラフィ専用計算回路

漸化式法を利用した CGH 計算回路を FPGA に実装した。振幅型の CGH を計算する専用計算回路のブロック図を図 5.8 に示した。専用計算回路は Point Cloud RAM, Recurrence Relation Unit (RRU) , RRU controller で構成した。

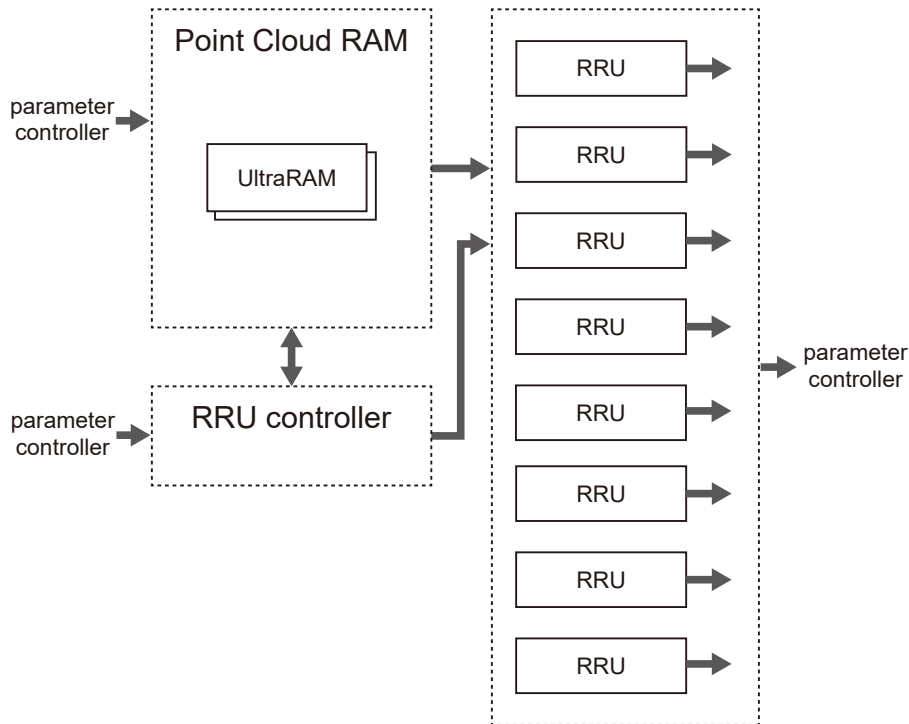


図 5.8: 振幅型電子ホログラフィ専用計算回路のブロック図.

Point Cloud RAMは、点群データをFPGA内部に保存するユニットである。FPGA内部のメモリブロックであるUltraRAMで構成した。図5.3に示した、parameter controllerから入力される点群用アドレスに従って、点群データをUltraRAMへと保存する。点群用アドレスはparameter controllerで0にオフセットしたアドレスであるため、別にmemory controllerを実装せず、そのままRAMに入力可能である。

RRUは、漸化式法によりCGHの画素値を計算するユニットである。本章では、振幅型のCGH計算回路として、8つのRRUが搭載されており、それぞれCGHの横1ライン分である1,920画素を計算する。8つのRRUで15,360画素を一度に計算可能である。RRU controllerは、parameter controllerから入力されるパラメータに応じ、RRUユニットが計算するCGHの座標を計算する。また、Point Cloud RAM内部のBlock RAMから点群データを読み込み、RRUでCGHを計算する。計算結果は、一時保存用のCGH RAM(図5.3)へと保存する。

ここで、複数のRRUにより並列計算を行うが、ZynqMPにあったRRU自体の並列数制限は改良により無くした。本章のシステムは、RRUが一度に並列計算可能な数は、自由に設定可能である。ただし、最も効率良くなる並列数は、計算したいCGHのサイズと同じか約数である。従って、本章では1つのRRUが並列計算する数は、 $1,920 \times 1,080$ 画素のCGH向けに、1,920画素とした。3,840 × 2,160画素といった

4K の CGH も回路構成を変更せずに計算することが可能である。ただし、CGH を一時保存している UltraRAM の容量を超えた CGH の計算はできない。現状ではシステム全体で最大 $3,840 \times 2,160$ 画素の CGH が計算可能である。専用計算機で計算を繰り返し、PC 側で画像を合成することで $3,840 \times 2,160$ 画素を超えた CGH も計算可能である。複数回計算を繰り返す場合は、データ通信のため、1 回あたりの計算時間が 0.1 ms 程増加する。

5.3.1 漸化式計算ユニットの実装

RRU は、漸化式法により CGH の計算を実行するユニットである。漸化式法を CGH の x 軸方向に関して適用し、一度に 1,920 画素の計算を行う。漸化式法における初期位相を計算する Basic Phase Unit (BPU)、加算位相を計算する Additional Phase Unit (APU)、光強度を計算する Intensity Unit (IU) から成り立つ。ZynqMP では、IU は BPU と APU の内部にそれぞれ実装されていた。U250 で RRU を実装し直す際、IU は BPU と APU の外部に実装した。BPU と APU の外に IU を出した構成の方が、将来的に点光源の振幅値の計算を加える場合 (本研究では 1 に固定して計算) など、拡張が容易である利点がある。また、BPU、APU をそのまま位相型の回路でも利用可能であり、IP としての再利用性が高まる。ブロック図を図 5.9 に示した。

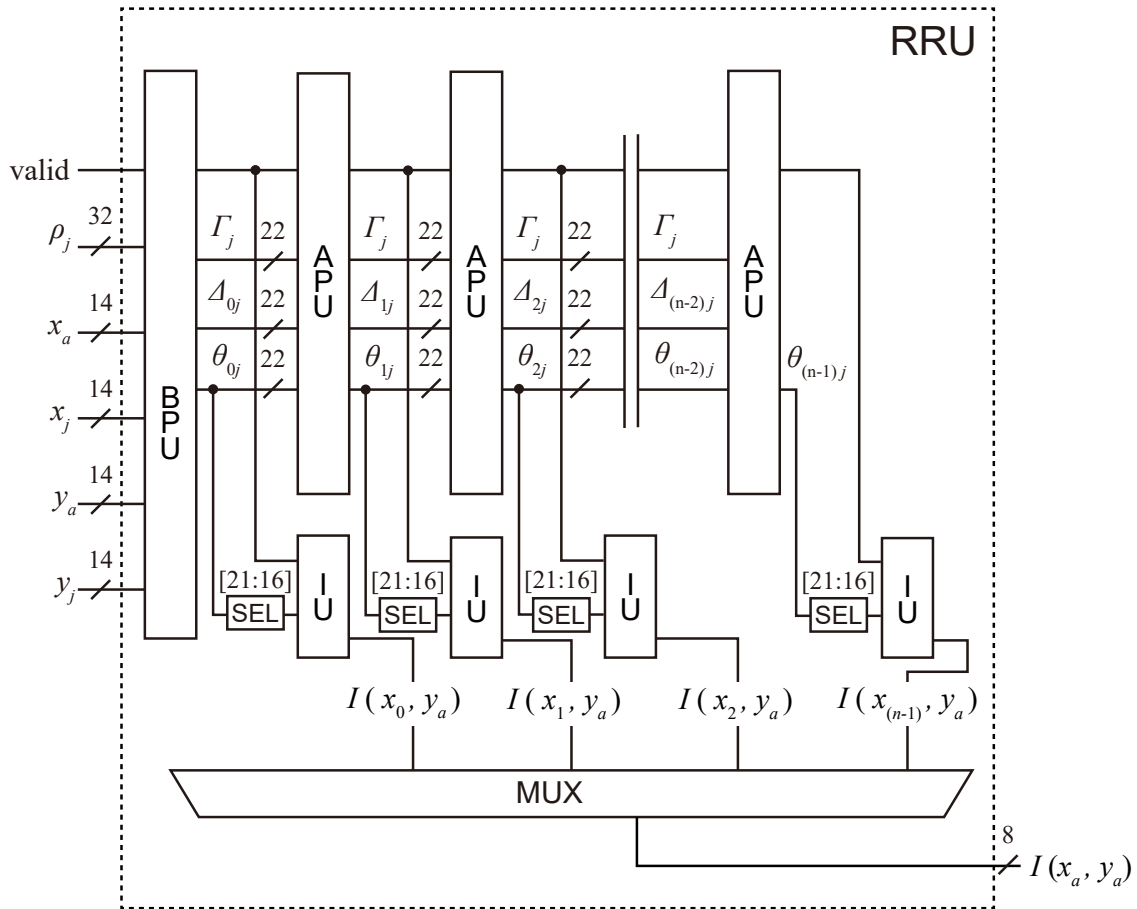


図 5.9: 漸化式計算部のブロック図. SEL は Selector を表す.

RRU は, CGH の座標値である x_a, y_a と点群データである x_j, y_j, ρ_j を入力として CGH 計算を行う. ここで, ρ_j は, 式 (1.17) に示す, $\rho_j = 1/2\lambda |z_j|$ である. valid シグナルは RRU コントローラによって生成される制御信号である. valid は有効な座標値と点群データが RRU に入力されているとき 1 となる. ZynqMP の RRU と異なり, 改良により, 計算すべき最後の点群データを表していた last シグナルは削減した. BPU, APU で計算された θ の上位 6 bit が IU へと入力され, 制御信号 valid の値により, 式 (1.18) に示した θ の総和計算を実行する. 計算結果の CGH 画素値は MultipleXer (MUX) により, ひとつひとつ出力される.

5.3.2 初期位相計算ユニットの実装

初期位相計算ユニットである BPU は, 漸化式法における初期位相を計算するユニットである. BPU 内部の計算回路を図 5.10 に示す. ZynqMP と同じく, CGH の座標値である x_a, y_a と点群データである x_j, y_j, ρ_j から, 式 (1.17), 式 (1.22), 式 (1.21)

の計算をする．ここでは ZynqMP と異なり，IU を BPU の外部へ実装したため，出力が θ ， Δ ， Γ ，valid のシンプルな構成となっている．また，last シグナルは改良により削減した．

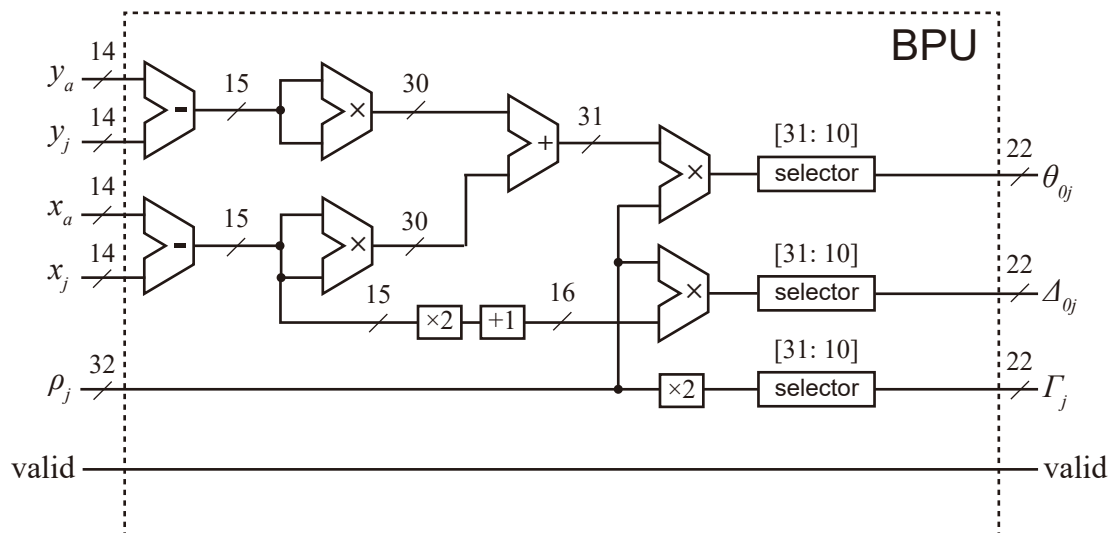


図 5.10: 初期位相計算部のブロック図.

5.3.3 加算位相計算ユニットの実装

加算位相計算ユニットである APU は，漸化式法における加算位相を計算するユニットである．APU 内部の計算回路を図 5.11 に示す．APU は，前段の BPU もしくは APU で計算した $\theta_{(n-2)j}$ ， $\Delta_{(n-2)j}$ ， Γ_j を入力として，式 (1.18)，式 (1.24)，式 (1.25) の計算を行う．こちらも BPU と同じく，IU を外部へ移し，last シグナルを削減した．

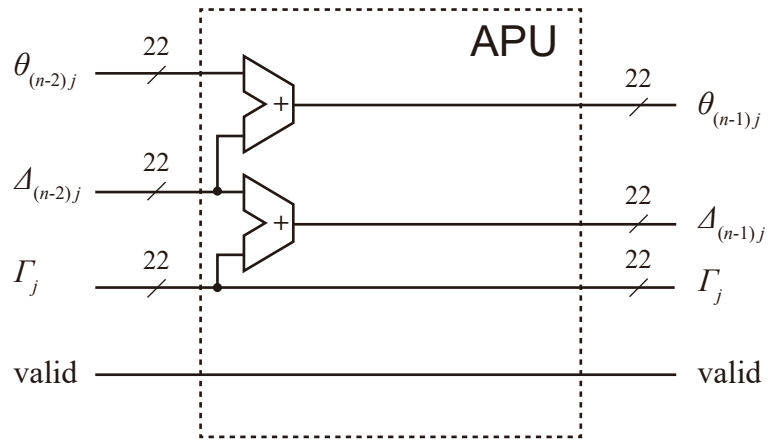


図 5.11: 加算位相計算部のブロック図.

5.3.4 光強度計算ユニットの実装

光強度計算ユニットである IU は、式 (1.18) の計算を行うユニットである。ブロック図を図 5.12 に示した。LUT の精度等、基本は ZynqMP の IU と同じである。ただし本章の IU は、制御信号 valid のみの値により、 $\theta_{(n-1)j}$ の余弦計算を行い、すべての点群に関して総和計算を実行する。

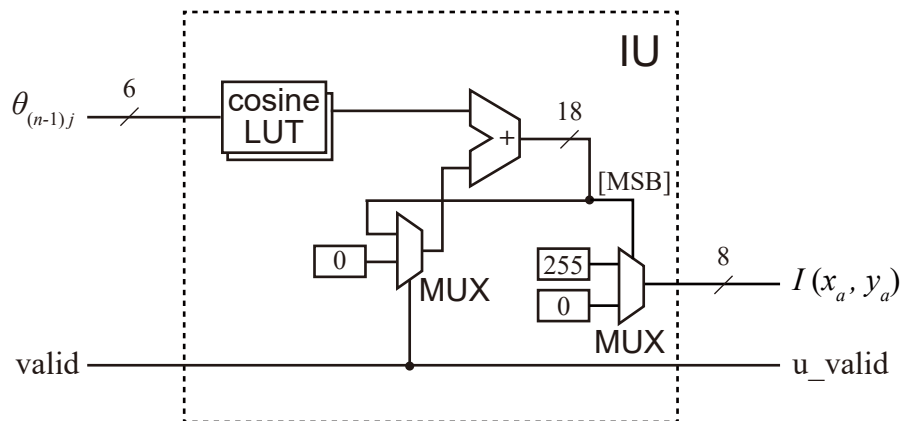


図 5.12: 光強度計算部のブロック図.

5.3.5 リソース使用率

U250 を用いて、専用通信回路と振幅型 CGH を計算する専用回路の実装を行った。CGH を計算する回路には 8 個の BPU と 15,352 個の APU を搭載した。一度に 15,360 画素を計算可能な専用計算機である。計算回路の動作周波数は 250 MHz である。FPGA リソースの使用状況を表 5.3 に示した。

表 5.3: 振幅型 CGH 専用計算機のリソース使用率。

Resource	Number of usage	Usage[%]
Look-up table	1,114,614	64.50
Flip flop	1,984,295	57.42
Block RAM	2.7 Mb	2.83
UltraRAM	40.5 Mb	11.25
DSP	40	0.33

5.3.6 計算性能評価

計算時間比較のため、CPU と GPU で CGH を計算するプログラムを作成した。実測した結果を図 5.13 に示す。CPU は Intel Core i9-9900K, Ubuntu 18.04.4 LTS (Linux Kernel 5.3.0-46-generic) が動作するシステムで、Intel C compiler 19.0.5.281 を用いた。CPU では、式 (1.18) のフレネル近似式と漸化式法を float 精度と int 精度でそれぞれ実装し比較した。また、CGH の高速化アルゴリズムとして代表的な N-LUT 法 [75] との比較も行った。GPU は NVIDIA RTX 2080 Ti を利用し、式 (1.18) のフレネル近似式と漸化式法を利用した CGH 計算プログラムを CUDA 10.2 を用いて実装した。CPU, GPU とともにすべての計算コアを利用し、並列に実行した。

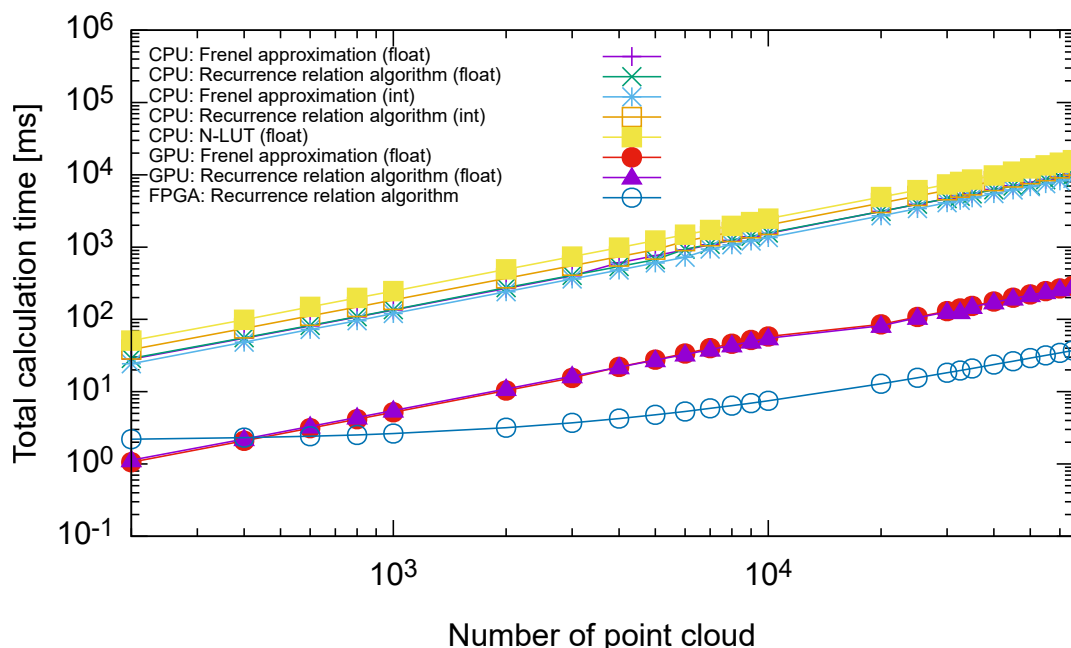


図 5.13: 計算時間の比較.

図 5.13 より, CPU では int 精度で実装したフレネル近似式が最も高速であった. GPU では, 僅差ではあるが, 漸化式法での実装が高速であった. 点群数が 50,000 点での CPU と GPU それぞれ最速の計算時間と U250 での計算時間の比較を表 5.4 に示した.

表 5.4: 振幅型 CGH 計算時間の比較.

Hardware	Calculation time [ms]	fps
FPGA (U250)	29	35
GPU	218	4.59
CPU	6,863	0.15

図 5.13 において, CPU や GPU と比べ U250 は線形なグラフになっていないことがわかる. これは, 専用計算機の実装によるものである. 本章で開発した専用計算機において, それぞれの RRU は完全に並列に動作する. 一方で, RRU 内部の BPU と APU はパイプライン実行するため, 完全に並列に動作せず, パイプラインディレイと呼ばれる遅延時間がある. 本章の実装では, 計算する点群数が BPU と APU の総数よりも小さい場合は, 計算コアを十分に活用できず, 性能が出ない仕様となっている. 計算する点群数が BPU と APU の総数よりも大きい場合 (ここでは数万点の点群を入力した場合), 並列動作とみなすことができる.

U250 における CGH 1 フレームあたりの理論的な計算時間は、パイプラインディレイを無視した場合、式 (5.1) で与えられる。

$$t_{U250} [s] = M \times K \div 15,360 \text{ 並列} \div 250 \text{ MHz} \quad (5.1)$$

ここで M は点群数、 K はホログラムの画素数を示す。点群数を 5 万点、ホログラムサイズを 200 万画素とすると、30 ms 秒程度であり、ほぼ理論値通りの結果が得られていることがわかる。表 5.4 より、開発した専用計算機は CPU と比べ 230 倍、GPU と比べ 7.5 倍の高速化を達成した。

5.4 位相型電子ホログラフィ専用計算回路

漸化式法を利用した CGH 計算回路を FPGA に実装した。位相型の CGH を計算する専用計算回路のブロック図を図 5.14 に示した。専用計算回路は Point Cloud RAM, Recurrence Relation Unit (RRU), RRU controller で構成した。

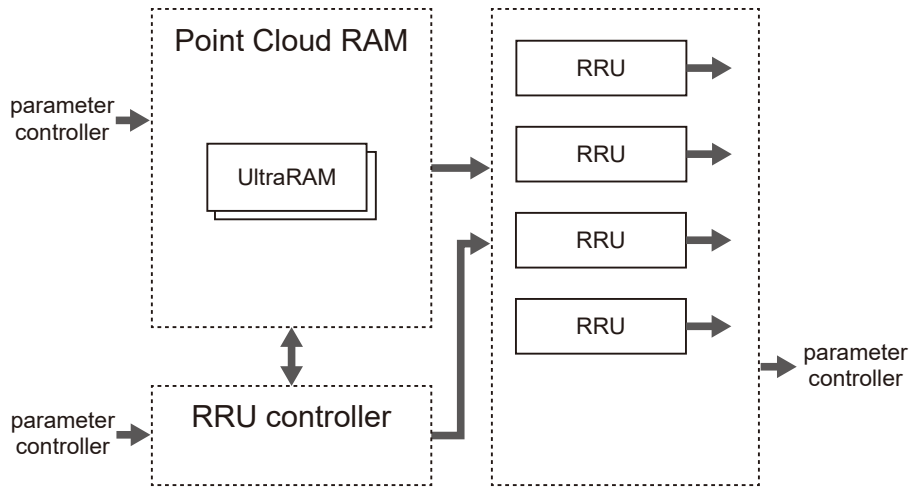


図 5.14: 位相型電子ホログラフィ専用計算回路のブロック図。

図 5.14 より、それぞれの役割は、RRU は位相型 CGH の計算を行うことを除いて、振幅型と同様である。RRU は、漸化式法により CGH の画素値を計算するが、計算に必要なリソースが増加しており、4 つの RRU のみ搭載した。4 つの RRU で 15,360 画素を一度に計算可能である。

5.4.1 漸化式計算ユニットの実装

漸化式計算ユニットである RRU は、漸化式法により CGH の計算を実行するユニットである。漸化式法を CGH の x 軸方向に関して適用し、一度に 1,920 画素の計算を行う。漸化式法における初期位相を計算する Basic Phase Unit (BPU)、加算位相を計算する Additional Phase Unit (APU)、複素振幅を計算する Complex amplitude Unit (CAU)、複素振幅分布の正規化を行う Normalization Unit (NU)、偏角計算用 LUT である arctan LUT から成り立つ。振幅型で利用した BPU と APU をそのまま再利用するために CAU は、ZynqMP の位相型 RRU と異なり BPU と APU の外部に実装した。

RRU のブロック図を図 5.15 に示した。RRU は、CGH の座標値である x_a, y_a と点群データである x_j, y_j, ρ_j を入力として複素振幅分布の計算を行う。振幅型同様、 $\rho_j = 1/2\lambda|z_j|$ は組み込み CPU を用いて計算を行い、リソースの増加や動作周波数の低下を抑えた。

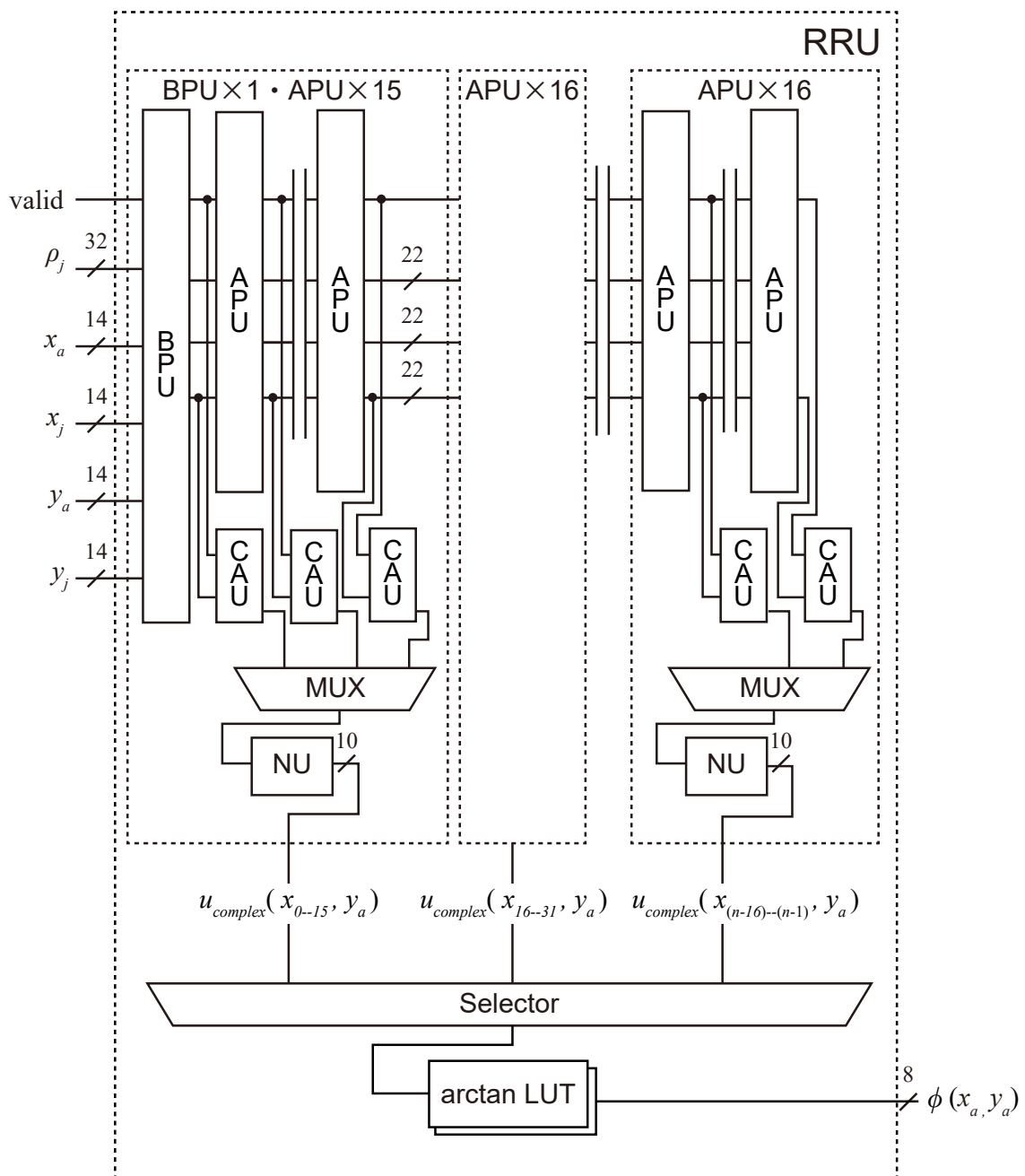


図 5.15: 漸化式計算部のブロック図.

valid シグナルは、RRU コントローラによって生成される制御信号である。valid は有効な座標値と点群データが RRU に入力されているとき、1 となる。ZynqMP の RRU と異なり、改良により計算すべき最後の点群データを表していた last シグナルは削減した。

ここで、ZynqMP のときと同じく、CAU に対して、NU, arctan LUT をどのよう

な比率で搭載するかという問題がある。本章では、必要なメモリサイズ、FPGAのタイミング制約等から比率の再検討を行ったが、大きな変化はなく、ZynqMPと同じく16並列ごとにNUを1つ搭載し、arctan LUTはRRUに対し1つだけ搭載した。

5.4.2 複素振幅計算ユニットの実装

複素振幅計算ユニットであるCAUは、式(1.19)内のcosとsinの総和計算を行うユニットである。ブロック図を図5.16に示した。LUTの精度等、基本はZynqMPと同じであるが、制御信号のlastシグナルを削減し、validの値のみにより、 $\theta_{(n-1)j}$ の余弦、正弦計算を行い、すべての点群に関して総和計算を実行する。

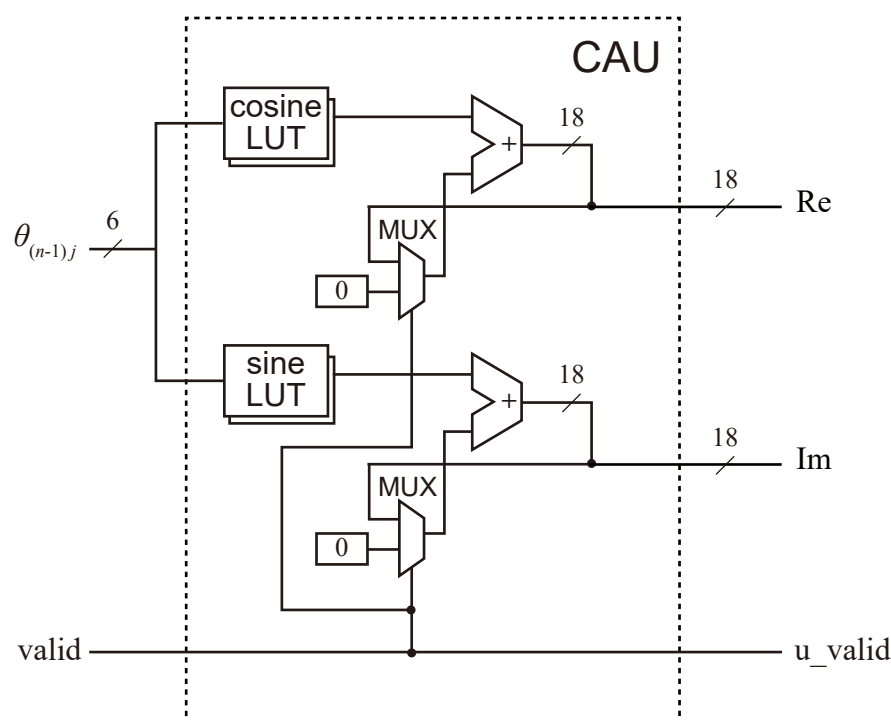


図 5.16: 複素振幅分布計算部のブロック図.

5.4.3 リソース使用率

U250を用いて、専用通信回路と位相型CGHを計算する専用回路の実装を行った。CGHを計算する回路には4個のBPUと7,676個のAPUを搭載した。一度に7,680画素を計算可能な専用計算機である。計算回路の動作周波数は250 MHzである。FPGAリソースの使用状況を表5.5に示した。

表 5.5: 位相型 CGH 専用計算機のリソース使用率.

Resource	Number of usage	Usage[%]
Look-up table	559,936	32.40
Flip flop	890,388	25.76
Block RAM	2.7 Mb	2.90
UltraRAM	22.5	6.25
DSP	7,937	64.59

5.4.4 計算性能評価

計算時間比較のため, CPU と GPU で CGH を計算するプログラムを作成した. 実測した結果を図 5.17 に示す. CPU は Intel Core i9-9900K, Ubuntu 18.04.4 LTS (Linux Kernel 5.3.0-46-generic) が動作するシステムで, Intel C compiler 19.0.5.281 を用いた. CPU では, 式 (1.19) のフレネル近似式と漸化式法を float 精度と int 精度でそれぞれ実装し比較した. また, CGH の高速化アルゴリズムとして代表的な N-LUT [75] との比較も行った. GPU は NVIDIA RTX 2080 Ti を利用し, 式 (1.19) のフレネル近似式と漸化式法を利用した CGH 計算プログラムを CUDA 10.2 を用いて実装した. CPU, GPU とともにすべての計算コアを利用し, 並列に実行した.

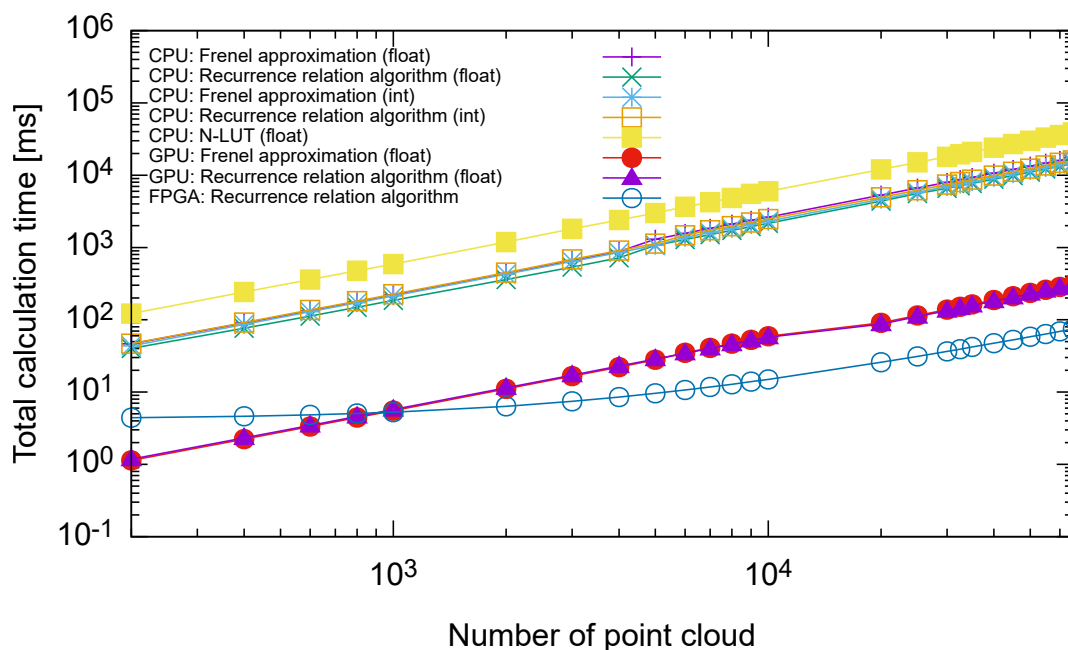


図 5.17: 計算時間の比較.

図 5.17 より，CPU では float 精度で実装した漸化式法が最も高速であった。GPU では，僅差ではあるが，漸化式法での実装が高速であった。点群数が 50,000 点での CPU と GPU それぞれ最速の計算時間と U250 での計算時間の比較を表 5.6 に示した。

表 5.6: 位相型 CGH 計算時間の比較.

Hardware	Calculation time [ms]	fps
FPGA (U250)	58	17
GPU	230	4.35
CPU	11,014	0.09

図 5.17 において，CPU や GPU と比べ U250 は線形なグラフになっていないことがわかる。これは，振幅型同様，専用計算機の実装によるものである。U250 における CGH 1 フレームあたりの理論的な計算時間は，パイプラインディレイを無視した場合，式 (5.2) で与えられる。

$$t_{U250} [s] = M \times K \div 7,680 \text{ 並列} \div 250 \text{ MHz} \quad (5.2)$$

ここで M は点群数， K はホログラムの画素数を示す。点群数を 5 万点，ホログラムサイズを 200 万画素とすると，60 ms 程度であり，ほぼ理論値通りの結果が得られていることがわかる。表 5.6 より，開発した専用計算機は CPU と比べ 190 倍，GPU と比べ 4 倍の高速化を達成した。

5.5 小括

本章では，振幅型と位相型の CGH を計算する電子ホログラフィ専用計算機の開発を行った。データセンター向けの大規模 FPGA ボードである Xilinx Alveo U250 データセンターアクセラレータカードを利用して，ワンチップで数千～数万並列規模の開発を行った。

FPGA に構築した回路には，AXI プロトコルに対応する通信回路と漸化式法を利用した CGH 計算回路を実装した。振幅型専用回路は，250 MHz で動作する 15,360 の専用計算コアにより，50,000 点の点群から $1,920 \times 1,080$ 画素の CGH を 35 fps で計算可能である。Intel Core i9-9900K (デスクトップ PC) と比べ，230 倍の高速化を達成した。位相型専用回路は，255 MHz で動作する 7,680 の専用計算コアにより，50,000 点の点群から $1,920 \times 1,080$ 画素の CGH を 17 fps で計算可能である。Intel Core i9-9900K (デスクトップ PC) と比べ，190 倍の高速化を達成した。

CGH 専用計算機としては，GPU を利用した高速化が研究されている。本章のシステムは NVIDIA RTX 2080 Ti と比べ，振幅型専用計算機は 7.5 倍，位相型専用計

算機は4倍の高速化を達成し、電子ホログラフィ計算での有用性を示した。また、先行研究のHORN-8システム [35,36] とチップ比較で、最大24倍の高速化を達成した。

本章のシステムでの計算精度は、4章のシステムと全く同じ精度となる。各演算のbit数やBPUとAPUの比率が計算精度へ影響を与える。本章のシステムでは、4章で実装したアーキテクチャから計算フローの改良や並列数の向上を行ったが、精度への影響はない。従って、再生像も同じになるため光学再生像に関しては省略した。

1,920 × 1,080画素のSLMでは、一度に表示可能な点光源数は5万点程度が上限である。振幅型のシステムは、地上デジタル放送のフレームレートである29.97 fpsでも十分に利用できる性能を達成した。位相型のシステムは、回路規模の増加により性能が半分になっているが、十分動画として表示可能な性能を達成した。

今後の展望として、U250ボードを複数台利用した、クラスタシステムなどにより、さらなる高速化を目指していきたい。時空間分割法 [71] を利用することで数十万点規模の表示が可能となる。また、SLMの大画面化も進んでる。さらなる高速化を達成することで、より高精細な3次元映像表示を実現したい。

結論

本研究では、計算の超高速化を実現することにより、電子・デジタルホログラフィ実用化への指針を示すことを目的とし、回路を再構成可能な Field Programmable Gate Array を用いて高速な計算機システムの開発を行った。

デジタルホログラフィ専用計算機の開発では、ハイスピードカメラを用いた小さい画素数 (128×128 画素) 向けの計算機を開発した。ハイスピードカメラでは、一般的に撮影フレームレートが増える程、画素数は小さくなる。画素数が小さい場合は、通常の計算機では並列化による計算効率が下がり、並列化による高速化が期待できない。一方で、専用計算機では、通常のシステムでは高速化が期待できないデータ数であっても高速に計算可能であることを確認した。

電子ホログラフィ専用計算機の開発では、専用計算回路の集積度を向上し、チップレベルでの並列数を向上した。最終的に CPU と FPGA 単体比較で、200 倍以上の高速化に成功した。 $1,920 \times 1,080$ の SLM で 1 度に表示可能な上限近い 5 万点の点光源を振幅型のシステムでは、地上デジタル放送のフレームレートである 29.97 fps でも十分に利用できる性能を達成した。位相型のシステムは、回路規模の増加により性能が半分になっているが、十分動画として表示可能な性能を達成した。

本論文の専用計算機を組み合わせることで、電子ホログラフィでは、観察者に負担をかけない Head Mounted Display を実現可能である。デジタルホログラフィでは、顕微鏡サイズで持ち運び可能なデジタルホログラフィ顕微鏡が実現可能となる。

今後の展望として、デジタルホログラフィ向け専用計算機では、より高精細な画像解析のため、計算アルゴリズムの改良が考えられる。電子ホログラフィでは、 $1,920 \times 1,080$ 画素のディスプレイ向けとして、漸化式法を利用した現状のアーキテクチャに大きな変更は必要無いと考えられる。ただし、位相型 CGH を計算するアルゴリズムには検討が必要である。本研究における単純な実装では、回路構成に必要な FPGA リソースが増加し、振幅型のシステムと比べ位相型では性能が半分になった。位相型 CGH を計算する代替アルゴリズムの検討が必要である。

参考文献

- [1] D. Gabor, “A new microscopic principle,” *Nature*, vol. 161, pp. 777–778, 1948.
- [2] 館暲, 佐藤誠, 廣瀬通孝, バーチャルリアリティ学 日本バーチャルリアリティ学会, 2014.
- [3] D. M. Hoffman, A. R. Girshick, K. Akeley, and M. S. Banks, “Vergence–accommodation conflicts hinder visual performance and cause visual fatigue,” *Journal of Vision*, vol. 8, no. 3, pp. 33–33, 2008.
- [4] M. Lambooi, M. Fortuin, I. Heynderickx, and W. IJsselstein, “Visual discomfort and visual fatigue of stereoscopic displays: A review,” *Journal of Imaging Science and Technology*, vol. 53, no. 3, pp. 30 201–1–30 201–14, 2009.
- [5] 不二門 尚, “小児の両眼視と 3D”, 日本視能訓練士協会誌, vol. 41, pp. 19–25, 2012.
- [6] P. St-Hilaire, S. A. Benton, M. E. Lucente, M. L. Jepsen, J. Kollin, H. Yoshikawa, and J. S. Underkoffler, “Electronic display system for computational holography,” in *Practical Holography IV*, S. A. Benton, Ed., vol. 1212, International Society for Optics and Photonics. SPIE, 1990, pp. 174–182.
- [7] J. W. Goodman, and R. W. Lawrence, “Digital image formation from electronically detected holograms,” *Applied Physics Letters*, vol. 11, no. 3, pp. 77–79, 1967.
- [8] S. Reichelt, R. Häussler, G. Fütterer, and N. Leister, “Depth cues in human visual perception and their realization in 3D displays,” in *Three-Dimensional Imaging, Visualization, and Display 2010 and Display Technologies and Applications for Defense, Security, and Avionics IV*, B. Javidi, J.-Y. Son, J. T. Thomas, and D. D. Desjardins, Eds., vol. 7690, International Society for Optics and Photonics. SPIE, 2010, pp. 92–103.

- [9] D. H. Barnhart, R. J. Adrian, and G. C. Papen, “Phase-conjugate holographic system for high-resolution particle-image velocimetry,” *Applied Optics*, vol. 33, no. 30, pp. 7159–7170, 1994.
- [10] H. Meng, and F. Hussain, “In-line recording and off-axis viewing technique for holographic particle velocimetry,” *Applied Optics*, vol. 34, no. 11, pp. 1827–1840, 1995.
- [11] J. Sheng, E. Malkiel, and J. Katz, “Single beam two-views holographic particle image velocimetry,” *Applied Optics*, vol. 42, no. 2, pp. 235–250, 2003.
- [12] U. Schnars, and W. Jüptner, “Direct recording of holograms by a CCD target and numerical reconstruction,” *Applied Optics*, vol. 33, no. 2, pp. 179–181, 1994.
- [13] S. Murata, and N. Yasuda, “Potential of digital holography in particle measurement,” *Optics & Laser Technology*, vol. 32, no. 7-8, pp. 567–574, 2000.
- [14] T. Kakue, R. Yonesaka, T. Tahara, Y. Awatsuji, K. Nishio, S. Ura, T. Kubota, and O. Matoba, “High-speed phase imaging by parallel phase-shifting digital holography,” *Optics Letters*, vol. 36, no. 21, pp. 4131–4133, 2011.
- [15] M. Lucente, “Interactive three-dimensional holographic displays: seeing the future in depth,” *ACM Siggraph Computer Graphics*, vol. 31, no. 2, pp. 63–67, 1997.
- [16] Y. Abe, N. Masuda, H. Wakabayashi, Y. Kazo, T. Ito, S.-i. Satake, T. Kunugi, and K. Sato, “Special purpose computer system for flow visualization using holography technology,” *Optics Express*, vol. 16, no. 11, pp. 7686–7692, 2008.
- [17] N. Masuda, T. Sugie, T. Ito, S. Tanaka, Y. Hamada, S.-i. Satake, T. Kunugi, and K. Sato, “Special purpose computer system with highly parallel pipelines for flow visualization using holography technology,” *Computer Physics Communications*, vol. 181, no. 12, pp. 1986–1989, 2010.
- [18] M. E. Lucente, “Interactive computation of holograms using a look-up table,” *Journal of Electronic Imaging*, vol. 2, no. 1, pp. 28–35, 1993.
- [19] K. Matsushima, and M. Takai, “Recurrence formulas for fast creation of synthetic three-dimensional holograms,” *Applied Optics*, vol. 39, no. 35, pp. 6587–6594, 2000.

- [20] T. Shimobaba, and T. Ito, “An efficient computational method suitable for hardware of computer-generated hologram with phase computation by addition,” *Computer Physics Communications*, vol. 138, no. 1, pp. 44–52, 2001.
- [21] H. Yoshikawa, “Fast computation of Fresnel holograms employing difference,” *Optical Review*, vol. 8, no. 5, pp. 331–335, 2001.
- [22] T. Shimobaba, N. Masuda, and T. Ito, “Simple and fast calculation algorithm for computer-generated hologram with wavefront recording plane,” *Optics Letters*, vol. 34, no. 20, pp. 3133–3135, 2009.
- [23] A. Symeonidou, D. Blinder, A. Munteanu, and P. Schelkens, “Computer-generated holograms by multiple wavefront recording plane method with occlusion culling,” *Optics Express*, vol. 23, no. 17, pp. 22 149–22 161, 2015.
- [24] Y. Ogihara, and Y. Sakamoto, “Fast calculation method of a CGH for a patch model using a point-based method,” *Applied Optics*, vol. 54, no. 1, pp. A76–A83, 2015.
- [25] S. Jiao, Z. Zhuang, and W. Zou, “Fast computer generated hologram calculation with a mini look-up table incorporated with radial symmetric interpolation,” *Optics Express*, vol. 25, no. 1, pp. 112–123, 2017.
- [26] D. Blinder, and P. Schelkens, “Accelerated computer generated holography using sparse bases in the STFT domain,” *Optics Express*, vol. 26, no. 2, pp. 1461–1473, 2018.
- [27] N. Masuda, T. Ito, T. Tanaka, A. Shiraki, and T. Sugie, “Computer generated holography using a graphics processing unit,” *Optics Express*, vol. 14, no. 2, pp. 603–608, 2006.
- [28] B. J. Jackin, S. Watanabe, K. Ootsu, T. Ohkawa, T. Yokota, Y. Hayasaki, T. Yatagai, and T. Baba, “Decomposition method for fast computation of gigapixel-sized Fresnel holograms on a graphics processing unit cluster,” *Applied Optics*, vol. 57, no. 12, pp. 3134–3145, 2018.
- [29] D.-W. Kim, Y.-H. Lee, and Y.-H. Seo, “High-speed computer-generated hologram based on resource optimization for block-based parallel processing,” *Applied optics*, vol. 57, no. 13, pp. 3511–3518, 2018.

- [30] H. Sannomiya, N. Takada, K. Suzuki, T. Sakaguchi, H. Nakayama, M. Oikawa, Y. Mori, T. Kakue, T. Shimobaba, and T. Ito, “Real-time spatiotemporal division multiplexing electroholography for 1,200,000 object points using multiple-graphics processing unit cluster,” *Chinese Optics Letters*, vol. 18, no. 7, p. 070901, 2020.
- [31] Y.-H. Seo, Y.-H. Lee, and D.-W. Kim, “ASIC chipset design to generate block-based complex holographic video,” *Applied Optics*, vol. 56, no. 9, pp. D52–D59, 2017.
- [32] P. W. M. Tsang, J. P. Liu, T. C. Poon, and K. W. K. Cheung, “Fast generation of hologram sub-lines based on field programmable gate array,” in *Digital Holography and Three-Dimensional Imaging*. Optical Society of America, 2009, p. DWC2.
- [33] Y.-H. Seo, H.-J. Choi, J.-S. Yoo, and D.-W. Kim, “An architecture of a high-speed digital hologram generator based on FPGA,” *Journal of Systems Architecture*, vol. 56, no. 1, pp. 27–37, 2010.
- [34] Y.-H. Seo, Y.-H. Lee, J.-S. Yoo, and D.-W. Kim, “Hardware architecture of high-performance digital hologram generator on the basis of a pixel-by-pixel calculation scheme,” *Applied optics*, vol. 51, no. 18, pp. 4003–4012, June 2012.
- [35] T. Sugie, T. Akamatsu, T. Nishitsuji, R. Hirayama, N. Masuda, H. Nakayama, Y. Ichihashi, A. Shiraki, M. Oikawa, N. Takada *et al.*, “High-performance parallel computing for next-generation holographic imaging,” *Nature Electronics*, vol. 1, no. 4, pp. 254–259, 2018.
- [36] T. Nishitsuji, Y. Yamamoto, T. Sugie, T. Akamatsu, R. Hirayama, H. Nakayama, T. Kakue, T. Shimobaba, and T. Ito, “Special-purpose computer HORN-8 for phase-type electro-holography,” *Optics Express*, vol. 26, no. 20, pp. 26 722–26 733, 2018.
- [37] H. Kim, Y. Kim, H. Ji, H. Park, J. An, H. Song, Y. T. Kim, H. Lee, and K. Kim, “A Single-Chip FPGA Holographic Video Processor,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 3, pp. 2066–2073, 2019.
- [38] T. Yabe, T. Ito, and M. Okazaki, “Holography Machine HORN-1 for Computer-Aided Retrieval of Virtual Three-Dimensional Image,” *Japanese Journal of Applied Physics*, vol. 32, no. Part 2, No. 9B, pp. L1359–L1361, Sep. 1993.

- [39] T. Ito, T. Yabe, M. Okazaki, and M. Yanagi, “Special-purpose computer HORN-1 for reconstruction of virtual image in three dimensions,” *Computer Physics Communications*, vol. 82, no. 2, pp. 104–110, 1994.
- [40] T. Ito, H. Eldeib, K. Yoshida, S. Takahashi, T. Yabe, and T. Kunugi, “Special-purpose computer for holography HORN-2,” *Computer Physics Communications*, vol. 93, no. 1, pp. 13–20, 1996.
- [41] Y. Yamamoto, N. Masuda, R. Hirayama, H. Nakayama, T. Kakue, T. Shimobaba, and T. Ito, “Special-purpose computer for electroholography in embedded systems,” *OSA Continuum*, vol. 2, no. 4, pp. 1166–1173, 2019.
- [42] 山本洋太, 増田信之, 中山弘敬, 下馬場朋禄, 角江崇, 伊藤智義, “System on a Chip を用いた小型な電子ホログラフィ専用計算機の開発”, *レーザー研究*, vol. 48, no. 10, pp. 555–559, 2020.
- [43] Y. Yamamoto, S. Namba, T. Kakue, T. Shimobaba, T. Ito, and N. Masuda, “Special-purpose computer for digital holographic high-speed three-dimensional imaging,” *Optical Engineering*, vol. 59, no. 5, p. 054105, 2020.
- [44] 伊藤智義, 下馬場朋禄, *ホログラフィ入門 コンピュータを利用した3次元映像・3次元計測* 株式会社 講談社, 2017.
- [45] T. Kakue, Y. Endo, T. Nishitsuji, T. Shimobaba, N. Masuda, and T. Ito, “Digital holographic high-speed 3D imaging for the vibrometry of fast-occurring phenomena,” *Scientific Reports*, vol. 7, no. 1, p. 10413, 2017.
- [46] K.-i. Aoshima, N. Funabashi, K. Machida, Y. Miyamoto, K. Kuga, T. Ishibashi, N. Shimidzu, and F. Sato, “Submicron magneto-optical spatial light modulation device for holographic displays driven by spin-polarized electrons,” *Journal of Display Technology*, vol. 6, no. 9, pp. 374–380, 2010.
- [47] Y. Isomae, Y. Shibata, T. Ishinabe, and H. Fujikake, “Design of 1- μm -pitch liquid crystal spatial light modulators having dielectric shield wall structure for holographic display with wide field of view,” *Optical Review*, vol. 24, no. 2, pp. 165–176, 2017.
- [48] K. Wakunami, P.-Y. Hsieh, R. Oi, T. Senoh, H. Sasaki, Y. Ichihashi, M. Okui, Y.-P. Huang, and K. Yamamoto, “Projection-type see-through holographic three-dimensional display,” *Nature communications*, vol. 7, no. 1, pp. 1–7, 2016.

- [49] R. G. Dorsch, A. W. Lohmann, and S. Sinzinger, “Fresnel ping-pong algorithm for two-plane computer-generated hologram display,” *Applied optics*, vol. 33, no. 5, pp. 869–875, 1994.
- [50] J. Amako, H. Miura, and T. Sonehara, “Speckle-noise reduction on kinoform reconstruction using a phase-only spatial light modulator,” *Applied Optics*, vol. 34, no. 17, pp. 3165–3171, 1995.
- [51] T. Shimobaba, and T. Ito, “Random phase-free computer-generated hologram,” *Optics Express*, vol. 23, no. 7, pp. 9549–9554, 2015.
- [52] ———, “Fast generation of computer-generated holograms using wavelet shrinkage,” *Optics Express*, vol. 25, no. 1, pp. 77–87, 2017.
- [53] K. Matsushima, M. Nakamura, and S. Nakahara, “Silhouette method for hidden surface removal in computer holography and its acceleration using the switch-back technique,” *Optics Express*, vol. 22, no. 20, pp. 24 450–24 465, 2014.
- [54] J.-P. Liu, and H.-K. Liao, “Fast occlusion processing for a polygon-based computer-generated hologram using the slice-by-slice silhouette method,” *Applied Optics*, vol. 57, no. 1, pp. A215–A221, 2018.
- [55] S. Igarashi, T. Nakamura, and M. Yamaguchi, “Fast method of calculating a photorealistic hologram based on orthographic ray–wavefront conversion,” *Optics Letters*, vol. 41, no. 7, pp. 1396–1399, 2016.
- [56] H. Sato, T. Kakue, Y. Ichihashi, Y. Endo, K. Wakunami, R. Oi, K. Yamamoto, H. Nakayama, T. Shimobaba, and T. Ito, “Real-time colour hologram generation based on ray-sampling plane with multi-GPU acceleration,” *Scientific Rep.*, vol. 8, no. 1, pp. 1–10, 2018.
- [57] J.-S. Chen, and D. P. Chu, “Improved layer-based method for rapid hologram generation and real-time interactive holographic display applications,” *Optics Express*, vol. 23, no. 14, pp. 18 143–18 155, 2015.
- [58] H. Zhang, L. Cao, and G. Jin, “Computer-generated hologram with occlusion effect using layer-based processing,” *Applied Optics*, vol. 56, no. 13, pp. F138–F143, 2017.
- [59] L. B. Lesem, P. M. Hirsch, and J. A. Jordan, “The kinoform: a new wavefront reconstruction device,” *IBM Journal of Research and Development*, vol. 13, no. 2, pp. 150–155, 1969.

- [60] 上原寛樹, 平井大智, 加賀城宏毅, 市橋保之, 中山弘敬, 下馬場朋祿, 増田信之, 伊藤智義, “位相変調型液晶ディスプレイを用いた電子ホログラフィ”, 映像情報メディア学会誌, vol. 65, no. 3, pp. 403–406, 2011.
- [61] 早崎芳夫, 光学ライブラリー7 デジタルホログラフィ 株式会社 朝倉書店, 2016.
- [62] “7 Series FPGAs Integrated Block for PCI Express v3.2,” https://www.xilinx.com/support/documentation/ip_documentation/pcie_7x/v3_2/pg054-7series-pcie.pdf.
- [63] “Xilinx Product Specification LogiCORE IP Fast Fourier Transform v7.1,” https://www.xilinx.com/support/documentation/ip_documentation/xfft_ds260.pdf.
- [64] “cuFFT,” <https://developer.nvidia.com/cufft>.
- [65] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [66] N. Takada, M. Fujiwara, C. Ooi, Y. Maeda, H. Nakayama, T. Kakue, T. Shimobaba, and T. Ito, “High-speed 3-d electroholographic movie playback using a digital micromirror device,” *IEICE Transactions on Electronics*, vol. E100.C, no. 11, pp. 978–983, 2017.
- [67] T. Ito, and T. Shimobaba, “One-unit system for electroholography by use of a special-purpose computational chip with a high-resolution liquid-crystal display toward a three-dimensional television,” *Optics Express*, vol. 12, no. 9, pp. 1788–1793, 2004.
- [68] T. Ito, N. Masuda, K. Yoshimura, A. Shiraki, T. Shimobaba, and T. Sugie, “Special-purpose computer HORN-5 for a real-time electroholography,” *Opt. Express*, vol. 13, no. 6, pp. 1923–1932, Mar. 2005.
- [69] Y. Ichihashi, H. Nakayama, T. Ito, N. Masuda, T. Shimobaba, A. Shiraki, and T. Sugie, “HORN-6 special-purpose clustered computing system for electroholography,” *Opt. Express*, vol. 17, no. 16, pp. 13 895–13 903, Aug. 2009.
- [70] T. Shimobaba, S. Hishinuma, and T. Ito, “Special-purpose computer for holography HORN-4 with recurrence algorithm,” *Computer physics communications*, vol. 148, no. 2, pp. 160–170, 2002.

- [71] Y. Yamamoto, H. Nakayama, N. Takada, T. Nishitsuji, T. Sugie, T. Kakue, T. Shimobaba, and T. Ito, “Large-scale electroholography by HORN-8 from a point-cloud model with 400,000 points,” *Optics Express*, vol. 26, no. 26, pp. 34 259–34 265, 2018.
- [72] “MicroBlaze,” https://xilinx.com/support/documentation/sw_manuals_j/xilinx2020_1/ug984-vivado-microblaze-ref.pdf.
- [73] “PetaLinux,” https://xilinx.com/support/documentation/sw_manuals_j/xilinx2020_2/ug1144-petalinux-tools-reference-guide.pdf.
- [74] “AXI DMA,” https://xilinx.com/support/documentation/ip_documentation/axi_dma/v7_1/pg021_axi_dma.pdf.
- [75] S.-C. Kim, and E.-S. Kim, “Effective generation of digital holograms of three-dimensional objects using a novel look-up table method,” *Applied Optics*, vol. 47, no. 19, pp. D55–D62, July 2008.
- [76] Y. Yamamoto, T. Shimobaba, H. Nakayama, T. Kakue, N. Masuda, and T. Ito, “System-on-a-chip-based special-purpose computer for phase electroholography,” *OSA Continuum*, vol. 3, no. 12, pp. 3407–3415, 2020.
- [77] S. Yamada, T. Kakue, T. Shimobaba, and T. Ito, “Interactive holographic display based on finger gestures,” *Scientific reports*, vol. 8, no. 1, pp. 1–7, 2018.

謝辞

本研究を始めるにあたり，貴重な研究の機会と研究環境を頂いた千葉大学 伊藤智義教授に心から感謝いたします。研究において，貴重な研究の機会とご助言を頂いた千葉大学 下馬場朋禄教授に深く感謝致します。専用計算機開発に関して，様々なご助言を頂いた東京理科大学 増田信之教授に深く感謝申し上げます。光学系関連において，様々なご助言を頂いた千葉大学 角江崇助教に深く感謝申し上げます。本研究で用いた数多くの素晴らしいCGデータをご提供頂いた，国立天文台 中山弘敬様に深く感謝致します。研究を行うにあたり，研究室の皆様にも大変お世話になりました。そして博士後期課程を修了まで支えてくれた家族に深く感謝いたします。簡単にではございますが，以上をもって謝辞とさせていただきます。

業績リスト

査読付き論文 (筆頭)

- [1] Yota Yamamoto, Hirotaka Nakayama, Naoki Takada, Takashi Nishitsuji, Takashige Sugie, Takashi Kakue, Tomoyoshi Shimobaba, Tomoyoshi Ito, "Large-Scale Electroholography by HORN-8 from Point-Cloud Model with 400,000 Points," *Optics Express*, vol. 26, no. 26, pp. 34259–34265, 2018.
- [2] Yota Yamamoto, Nobuyuki Masuda, Ryuji Hirayama, Hirotaka Nakayama, Takashi Kakue, Tomoyoshi Shimobaba, Tomoyoshi Ito, "Special-purpose computer for electroholography in embedded systems," *OSA Continuum*, vol. 2, no. 4, pp. 1166–1173, 2019.
- [3] Yota Yamamoto, Shintaro Namba, Takashi Kakue, Tomoyoshi Shimobaba, Tomoyoshi Ito, Nobuyuki Masuda, "Special-purpose computer for digital holographic high-speed 3D imaging," *Optical Engineering*, vol. 59, no. 5, pp. 054105, 2020.
- [4] 山本洋太, 増田信之, 中山弘敬, 下馬場朋禄, 角江崇, 伊藤智義, "System on a Chip を用いた小型な電子ホログラフィ専用計算機の開発," *レーザー研究*, vol. 48, no. 10, pp. 555–559, 2020.
- [5] Yota Yamamoto, Tomoyoshi Shimobaba, Hirotaka Nakayama, Takashi Kakue, Nobuyuki Masuda, Tomoyoshi Ito, "System-on-a-chip-based special-purpose computer for phase electroholography," *OSA Continuum*, vol. 3, no. 12, pp. 3407–3415, 2020.

査読付き論文 (共著)

- [1] Takashi Nishitsuji, Yota Yamamoto, Takashige Sugie, Takanori Akamatsu, Ryuji Hirayama, Hirotaka Nakayama, Takashi Kakue, Tomoyoshi Shimobaba, Tomoyoshi Ito, "Special-purpose computer HORN-8 for phase-type electroholography," *Optics Express*, vol. 26, no. 20, pp. 26722–26733, 2018.

- [2] Tomoyoshi Shimobaba, Takayuki Takahashi, Yota Yamamoto, Takashi Nishitsuji, Atushi Shiraki, Takashi Kakue, Tomoyoshi Ito, "Efficient diffraction calculations using implicit convolution," *OSA Continuum*, vol. 1, no. 2, pp. 642–650, 2018.
- [3] Tomoyoshi Shimobaba, Takayuki Takahashi, Yota Yamamoto, Yutaka Endo, Atsushi Shiraki, Takashi Nishitsuji, Naoto Hoshikawa, Takashi Kakue, Tomoyoshi Ito, "Digital holographic particle volume reconstruction using a deep neural network," *Applied Optics*, vol. 58, no. 8, pp. 1900–1906, 2019.
- [4] Satoki Hasegawa, Hidenari Yanagihara, Yota Yamamoto, Takashi Kakue, Tomoyoshi Shimobaba, Tomoyoshi Ito, "Electroholography of real scenes by RGB-D camera and the downsampling method," *OSA Continuum*, vol. 2, no. 5, pp. 1629–1638, 2019.
- [5] Hidenari Yanagihara, Takashi Kakue, Yota Yamamoto, Tomoyoshi Shimobaba, Tomoyoshi Ito, "Real-time three-dimensional video reconstruction of real scenes with deep depth using electro-holographic display system," *Optics Express*, vol. 27, no. 11, pp. 15662–15678, 2019.
- [6] Tomoyoshi Shimobaba, David Blinder, Michal Makowski, Peter Schelkens, Yota Yamamoto, Ikuo Hoshi, Takashi Nishitsuji, Yutaka Endo, Takashi Kakue, Tomoyoshi Ito, "Dynamic range compression scheme for digital hologram using a deep neural network," *Optics Letters*, vol. 44, no. 12, pp. 3038–3041, 2019.
- [7] Yoshiya Wagatsuma, Tomoyoshi Shimobaba, Yota Yamamoto, Ikuo Hoshi, Takashi Kakue, Tomoyoshi Ito, "Phase retrieval using axial diffracted patterns and ptychographic iterative engine," *Applied Optics*, vol. 59, no. 2, pp. 354–362, 2020.
- [8] Tomoyoshi Shimobaba, Michal Makowski, Takayuki Takahashi, Yota Yamamoto, Ikuo Hoshi, Takashi Kakue, Tomoyoshi Ito, "Reducing computational complexity and memory usage of iterative hologram optimization using scaled diffraction," *Applied Sciences*, vol. 10, no. 3, pp. 1132–1143, 2020.
- [9] Tomoyoshi Shimobaba, Takayuki Takahashi, Yota Yamamoto, Ikuo Hoshi, Atsushi Shiraki, Takashi Kakue, Tomoyoshi Ito, "Simple complex amplitude encoding of a phase-only hologram using binarized amplitude," *Journal of Optics*, vol. 22, no. 4, pp. 045703, 2020.

- [10] Tomoyoshi Shimobaba, Takashi Kakue, Yota Yamamoto, Ikuo Hoshi, Harutaka Shiomi, Takashi Nishitsuji, Naoki Takada, Tomoyoshi Ito, "Hologram generation via Hilbert transform," *OSA Continuum*, vol. 3, no. 6, pp. 1498–1503, 2020.
- [11] Daiki Yasuki, David Blinder, Tomoyoshi Shimobaba, Yota Yamamoto, Ikuo Hoshi, Peter Schelkens, Takashi Kakue, Tomoyoshi Ito, "Dedicated hardware for hologram calculation using sparse Fourier bases," *Applied Optics*, vol. 59, no. 26, pp. 8029–8037, 2020.
- [12] 山口由美, 長浜佑樹, 山本洋太, 下馬場朋禄, 角江崇, 伊藤智義, "ランダム位相フリー法を用いた位相型ホログラムの画質改善とシミュレーション及び光学実験による検証," *映像情報メディア学会誌*, vol. 74, no. 2, pp. 390–394, 2020.

国際会議 (筆頭)

- [1] Yota Yamamoto, Nobuyuki Masuda, Takashi Kakue, Tomoyoshi Shimobaba, Tomoyoshi Ito, "Development of special purpose computer for holography on embedded system," *International Conference in 3D systems and Applications (3DSA2018)*, 3D6-3, Taiwan, 29-30 Aug. 2018.
- [2] Yota Yamamoto, Nobuyuki Masuda, Takashi Kakue, Nishitsuji Takashi, Tomoyoshi Shimobaba and Tomoyoshi Ito, "Special Purpose computer for holography on the embedded system," *IDW 18*, HT-6, Nagoya Congress Center, 12-14 Dec. 2018.
- [3] Yota Yamamoto, Nobuyuki Masuda, Hirotaka Nakayama, Tomoyoshi Shimobaba, Takashi Kakue, Tomoyoshi Ito, "Phase type special-purpose computer for electroholography in embedded systems," *8th International Congress on Advanced Applied Informatics*, EAIS-WSS 7-2, Toyama International Conference Center Japan, 7-12 Jul. 2019.
- [4] Yota Yamamoto, Nobuyuki Masuda, Hirotaka Nakayama, Tomoyoshi Shimobaba, Takashi Kakue, Tomoyoshi Ito, "Highly parallel special-purpose computer for electroholography on system on a chip," *International Conference in 3D systems and Applications (3DSA2019)*, 3Dp1/3DSAp1, Sapporo, 27-29 Nov. 2019.

- [5] Yota Yamamoto, Tomoyoshi Shimobaba, Hirotaka Nakayama, Takashi Kakue, Nobuyuki Masuda, Tomoyoshi Ito, "Highly Parallel Special-Purpose Computer for Amplitude Electroholography," *International Workshop on Holography and related technologies (IWH2020)*, Sun Moon Lake, Taiwan, 13-15 Oct. 2020.

国際会議 (共著)

- [1] Takashi Nishitsuji, Yota Yamamoto, Takashige Sugie, Hirotaka Nakayama, Takashi Kakue, Tomoyoshi Shimobaba, Tomoyoshi Ito, "Recent progress of dedicated computer for electro-holography," *The 8th international Symposium on Holography (HODIC in Taiwan 5)*, vol. 0925-0950, National Chiao Tung University, 13-15 Dec. 2018.
- [2] Takashi Nishitsuji, Yota Yamamoto, Takashige Sugie, Takashi Kakue, Hirotaka Nakayama, Tomoyoshi Shimobaba, and Tomoyoshi Ito, "Dedicated computer for computer holography and its future outlook," *SPIE Sensing Technology + Applications*, vol. 10997-16, Baltimore USA, 14-18 Apr. 2019.
- [3] Tomoyoshi Shimobaba, David Blinder, Peter Schelkens, Yota Yamamoto, Ikuo Hoshi, Takashi Kakue and Tomoyoshi Ito, "Deep-learning-assisted hologram calculation via low-sampling holograms," *8th International Congress on Advanced Applied Informatics*, EAIS-WSS 7-3, Toyama International Conference Center Japan, 7-12 Jul. 2019.
- [4] Takashi Kakue, Hidenari Yanagihara, Yota Yamamoto, Tomoyoshi Shimobaba, Tomoyoshi Ito, "Demonstration of real-time holographic display system for three-dimensional reconstruction of real scenes", *HICO*, Gyeongju, Korea, 27-30 Aug. 2019.
- [5] Tomoyoshi Shimobaba, David Blinder, Peter Schelkens, Yota Yamamoto, Ikuo Hoshi, Atsushi Shiraki, Takashi Kakue, Tomoyoshi Ito, "Deep-learning-based dynamic range compression for 3D scene hologram", *International Conference on Optics & Electro-Optics XLIII Symposium of Optical Society of India (ICOL-2019)*, IRDE, Uttarakhand India, 19-22 Oct. 2019.
- [6] Yumi Yamaguchi, Yuki Nagahama, Yota Yamamoto, Tomoyoshi Shimobaba, Takashi Kakue, Tomoyoshi Ito, "Improvement of Image Quality of Phase-Only

Hologram using Random Phase-Free Method: Verification by Simulation and Optical Experiments,” *SPIE Photonics Europe*, France, 29 Mar. - 2 Apr. 2020.

- [7] Tomoyoshi Shimobaba, Yota Yamamoto, Ikuo Hoshi, Takashi Kakue, Tomoyoshi Ito, ”Holographic classification and regression using binary neural network,” *IEEE 18th International Conference on Industrial Informatics*, 2020.
- [8] Tomoyoshi Shimobaba, Yota Yamamoto, Takashi Nishitsuji, Ikuo Hoshi, Harutaka Shiomi, Takashi Kakue, Tomoyoshi Ito, ”Dedicated processor for holography assisted by deep neural networks,” *OSA Frontiers in Optics 2020 + Laser Science (FiO/LS)*, the Marriott Wardman Park in Washington, DC, USA, 13-17 Sep. 2020.

国内会議 (筆頭)

- [1] 山本洋太, 山形健太, 廣田祐輔, 増田信之, 原田哲也, “組み込みシステム向けホログラフィ専用計算機の開発”, C-009, 第 16 回情報科学技術フォーラム (FIT2017), 東京大学, 12-14 Sep. 2017.
- [2] 山本洋太, 杉江崇繁, 西辻崇, 下馬場朋禄, 角江崇, 伊藤智義, ”電子ホログラフィ専用計算機 HORN-8 を用いた 3 次元映像システム”, 日本光学会ホログラフィック・ディスプレイ研究会 (OSJ-HODIC), vol. 38, No. 3, 東京都調布市, 7 Sep. 2018.
- [3] 山本洋太, 増田信之, 杉江崇繁, 西辻崇, 角江崇, 下馬場朋禄, 伊藤智義, “次世代電子ホログラフィ専用計算機へ向けた計算回路の検討”, 第 19 回日本光学会情報フォトンクス研究グループ研究会 (秋合宿), 那須オオシマフォーラム, 栃木県, 9-10 Sep. 2018.
- [4] 山本洋太, 増田信之, 角江崇, 下馬場朋禄, 伊藤智義, “組み込みシステム向けホログラフィ専用計算機の開発”, C-003, 第 17 回情報科学技術フォーラム (FIT2018), 福岡工業大学, 19-21 Sep. 2018.
- [5] 山本洋太, 杉江崇繁, 角江崇, 下馬場朋禄, 伊藤智義, “電子ホログラフィ専用計算機の開発”, 千葉大学 グローバルプロミネント研究基幹シンポジウム (第 3 回), No.21, 千葉大学, 6 Nov. 2018.
- [6] 山本洋太, 増田信之, 角江崇, 下馬場朋禄, 伊藤智義, ”漸化式を用いた組み込み機器向け電子ホログラフィ専用計算機の開発”, 情報処理学会第 81 回全国大会, 1A-01, 福岡大学, 14-16 Mar. 2019.

- [7] 山本洋太, 増田信之, 角江崇, 下馬場朋禄, 伊藤智義, ”組み込み機器向けリアルタイム電子ホログラフィ専用計算機の開発”, 第 227 回システム・アーキテクチャ・第 187 回システムと LSI の設計技術・第 50 回組み込みシステム合同研究発表会 (ETNET2019), Vol.2019-SLDM-187 No.48, 鹿児島県西之表市西之表市民会館, 17-18 Mar. 2019.
- [8] 山本洋太, 増田信之, 下馬場朋禄, 角江崇, 伊藤智義, ”電子ホログラフィ向け専用計算機の開発”, LSI とシステムのワークショップ 2019, No.51, 東京大学生産技術研究所 総合研究実験棟 (An 棟) 2 階 コンベンションホール, 13-14 May 2019.
- [9] 山本洋太, 増田信之, 下馬場朋禄, 角江崇, 伊藤智義, ”組み込み機器向け電子ホログラフィ専用計算機の開発による次世代 3 次元映像システム”, 第 13 回新画像システム・情報フォトニクス研究討論会, B-6, キャンパスイノベーションセンター (田町 CIC), 6 Jun. 2019.
- [10] 山本洋太, 増田信之, 下馬場朋禄, 角江崇, 伊藤智義, ”組み込みシステム向け位相変調型電子ホログラフィ専用計算機の開発”, C-005, 第 18 回情報科学技術フォーラム (FIT2019), 岡山大学津島キャンパス, Sep. 3-5 2019.
- [11] 山本洋太, 増田信之, 下馬場朋禄, 角江崇, 伊藤智義, ”ホログラフィ専用計算機の開発による次世代映像システム”, 第 20 回情報フォトニクス研究グループ研究会 (秋合宿), 徳島県, 23-25 Sep. 2019.
- [12] 山本洋太, 増田信之, 下馬場朋禄, 角江崇, 伊藤智義, ”System on a Chip を用いた電子ホログラフィ向け超並列専用計算機の開発”, グローバルプロミネント研究基幹シンポジウム (第 4 回), 千葉大学, 5 Dec. 2019.
- [13] 山本洋太, 増田信之, 下馬場朋禄, 角江崇, 伊藤智義, ”System on a Chip を用いた小型な電子ホログラフィ専用計算機の開発”, H04-21p-XII-03, レーザー学会 学術講演会第 40 回年次大会, 仙台, 20-22 Jun. 2020.
- [14] 山本洋太, 増田信之, 下馬場朋禄, 角江崇, 伊藤智義, ”Xilinx Alveo U250 を用いた電子ホログラフィ専用計算機の開発”, CPSY-18, 組込み技術とネットワークに関するワークショップ ETNET2020, 鹿児島, 27-28 Feb. 2020.
- [15] 山本洋太, 増田信之, 下馬場朋禄, 角江崇, 伊藤智義, ”Xilinx Alveo U250 を用いた電子ホログラフィ専用計算機の実装”, RECONF2020-9, リコンフィギュラブルシステム研究会, オンライン, 28-29 May 2020.

- [16] 山本洋太, 増田信之, 下馬場朋禄, 角江崇, 伊藤智義, ”Xilinx Alveo U250 を用いた位相変調型電子ホログラフィ専用計算機の開発”, C-005, 第 19 回情報科学技術フォーラム (FIT2020), オンライン, 1-3 Sep. 2020.
- [17] 山本洋太, 下馬場朋禄, 角江崇, 増田信之, 伊藤智義, ”Xilinx Alveo U250 を用いた振幅・位相変調型ホログラフィ向け専用計算機の開発”, 令和 2 年第 3 回ホログラフィック・ディスプレイ研究会, オンライン 25 Sep. 2020.
- [18] 山本洋太, 下馬場朋禄, 角江崇, 増田信之, 伊藤智義, ”専用計算機による電子ホログラフィを用いた 3 次元映像システムの開発”, 日本光学会年次学術講演会 光みらい奨励金 記念講演, オンライン, 16 Nov. 2020.
- [19] 山本洋太, 下馬場朋禄, 角江崇, 増田信之, 伊藤智義, ”電子ホログラフィ専用計算機による 3 次元映像システムの開発”, グローバルプロミネント研究基幹シンポジウム (第 5 回), 千葉大学, 18 Nov. 2020.

国内会議 (共著)

- [1] 山形健太, 山本洋太, 廣田祐輔, 難波慎太郎, 増田信之, 阿比留慎, 角江崇, 下馬場朋禄, 伊藤智義, “三次元計測システムのためのデジタルホログラフィ専用計算機の開発”, C-010, 第 16 回情報科学技術フォーラム (FIT2017), 東京大学, 12-14 Sep. 2017.
- [2] 廣田祐輔, 山形健太, 山本洋太, 山本未来呂, 南昇吾, 増田信之, “パーシャルリコンフィグレーションを用いた専用計算機システムの開発”, C-011, 第 16 回情報科学技術フォーラム (FIT2017), 東京大学, 12-14 Sep 2017.
- [3] 西辻崇, 山本洋太, 杉江崇繁, 下馬場朋禄, 角江崇, 伊藤智義, ”ホログラフィ計算における三角関数近似手法の高精度化”, 日本光学会ホログラフィック・ディスプレイ研究会 (OSJ-HODIC), Vol. 38, No. 3, 東京都調布市, 7 Sep. 2018.
- [4] 佐藤紘将, 廣田祐輔, 山本洋太, 増田信之, ”高位合成を用いた 2 次元 FFT 計算回路の作成と検証”, C-002, 第 17 回情報科学技術フォーラム (FIT2018), 福岡工業大学, 19-21 Sep 2018.
- [5] 廣田祐輔, 佐藤紘将, 山本洋太, 増田信之, “動的部分再構成技術を用いた専用計算機システムの開発”, C-004, 第 17 回情報科学技術フォーラム (FIT2018), 福岡工業大学, 19-21 Sep 2018.

- [6] 増田信之, 廣田祐輔, 佐藤紘将, 山形健太, 山本洋太, 角江崇, 下馬場朋禄, 伊藤智義, “高速イメージング用専用計算機の開発”, C-005, 第17回情報科学技術フォーラム (FIT2018), 福岡工業大学, 19-21 Sep 2018.
- [7] 増田信之, 佐藤紘将, 祢津佑, 大森一世, 廣田祐輔, 山本洋太, 角江崇, 下馬場朋禄, 伊藤智義, ”FPGA を用いた高速イメージング用専用計算機システム”, C-006, 第18回情報科学技術フォーラム (FIT2019), 岡山大学津島キャンパス, 3-5 Sep. 2019.
- [8] 山口由美, 長浜佑樹, 山本洋太, 下馬場朋禄, 角江崇, 伊藤智義, ”ランダム位相フリー法を用いた位相型ホログラムの画質改善とシミュレーションによる検証”, 日本光学会ホログラフィック・ディスプレイ研究会 (OSJ-HODIC), 首都大学東京 秋葉原サテライトキャンパス, 13 Sep. 2019.
- [9] 渡邊峻介, 西辻崇, 山本洋太, 星郁雄, 干川尚人, 伊藤智義, ”FPGA を用いた避難行動予測・誘導デバイスの開発”, 110, 超知性ネットワークに関する分野横断型研究会 (RISING), 東京大学, 26-27 Nov. 2019.
- [10] 渡邊峻介, 西辻崇, 山本洋太, 星郁雄, 干川尚人, 伊藤智義, ”FPGA を用いた避難行動予測回路の設計”, A-19-4, 電子情報通信学会 総合大会, 広島大学, 17-20 Mar. 2020.
- [11] 山口由美, 長浜佑樹, 山本洋太, 角江崇, 伊藤智義, 下馬場朋禄, ”位相型ホログラフィックプロジェクトにおける画質改善アルゴリズム”, 令和2年第3回ホログラフィック・ディスプレイ研究会, オンライン, 25 Sep. 2020.

解説記事

- [1] 山田翔大, 山本洋太, 三瓶卓方, 伊藤智義, ”DFD 錯視を利用したスマートフォンによる小型立体映像装置の開発”, 画像ラボ, vol. 29, no. 11, pp. 33-39, Nov. 2018.
- [2] 下馬場朋禄, 星郁雄, 山本洋太, 塩見日隆, 角江崇, 伊藤智義, ”小型化を志向したホログラフィックプロジェクト”, 光アライアンス, vol. 31, no. 8, pp. 50-53, Aug. 2020.

外部資金

- [1] 千葉大学 国際交流事業若手先導人材育成支援プログラム, 100 千円, 2018.

- [2] European Commission EURAXESS Scholarship for PhD student in Diffractive Optics / Computer Holography, 72,000 PLN (約 2,000 千円), 2018-2019.
- [3] 千葉大学 国際交流事業若手先導人材育成支援プログラム, 252 千円, 2019.
- [4] コニカミノルタ光みらい若手奨励金, 300 千円, 2019.
- [5] 情報処理学会 大会奨励賞 (旧学術奨励賞), 10 千円, 2019.
- [6] 日本学術振興会 特別研究員奨励費, 1,100 千円, 2020.

受賞

- [1] 情報処理学会第 81 回全国大会 大会奨励賞 (旧学術奨励賞), Mar. 2019.
- [2] 第 18 回情報科学技術フォーラム FIT 奨励賞, Sep. 2019.
- [3] コニカミノルタ光みらい若手奨励金, Dec. 2019.
- [4] 第 19 回情報科学技術フォーラム FIT 奨励賞, Sep. 2020.
- [5] ホログラフィック・ディスプレイ研究会 学生優秀発表賞, Sep. 2020.