

# FPGA を用いたアルゴリズム実装 中間報告

電子情報工学科 3 年 太田智也

2025-10-09

## 課題 1 (ルーレット)

- 1-A: SW[1:0] に応じた速度で 7seg を回転させる
- 1-B: SW[2] の入力を追加し，これに応じて回転方向を変える

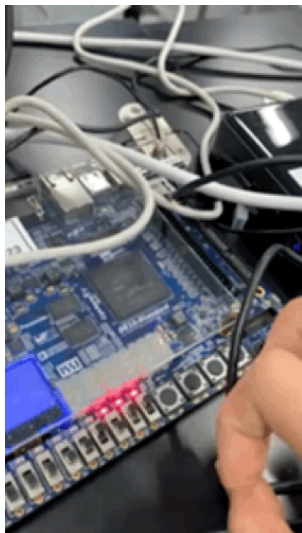


図 1: 課題 1 の動作例

```
1 end else if( cntout1 == 0 && cntout0 == 1 ) begin
2     if(dir == 0) begin
3         rr <= {rr[4:0],rr[5]};
4     end else begin
5         rr <= {rr[0],rr[5:1]};
6     end
```

リスト 1: 逆回転部分のコード

# 課題 1 (ルーレット)

- リソース・性能はあまり変化せず
- fanout は増加した

表 1: 課題 1 の使用リソース

	1-A	1-B
Logic usage	24	24
ALUT usage	43	43
Dedicated logic reg	39	39
IO pins	15	16
Max fanout	39	39
Total fanout	245	258
Ave fanout	2.19	2.26

表 2: 課題 1 の最大動作周波数(@1.1V, 0°C)

	1-A	1-B
CLOCK_50	389.71 MHz	388.95 MHz

## 課題 2 (リモコンと電卓)

- リモコンの信号を受信し, 四則演算の結果を 7seg に表示する
- どのリソースも課題 1 より大幅に増加し, DSP が 1 つ使用された

表 3: 課題 2 の使用リソース

	2
Logic usage	576
ALUT usage	1109
Dedicated logic reg	188
IO pins	69
Total DSP blocks	1
Max fanout	188
Total fanout	4406
Ave fanout	3.07

表 4: 課題 2 の最大動作周波数(@1.1V, 0°C)

	2
CLK	33.54 MHz

## 課題 3 (VGA による画面表示)

- 3-1: 座標に応じて RGB を変化させて模様を表示する
- 3-2: てきとうな図形(円)の描画
- 3-3: ASCII コードを用いた文字の描画

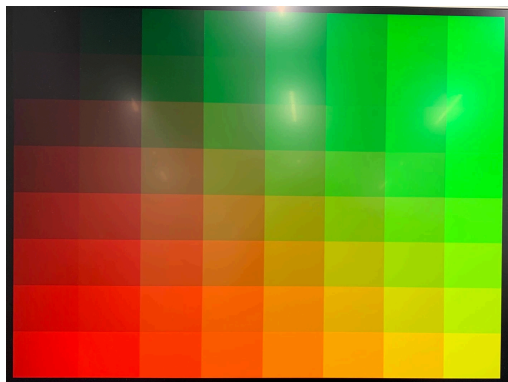


図 2: 3-1 の表示

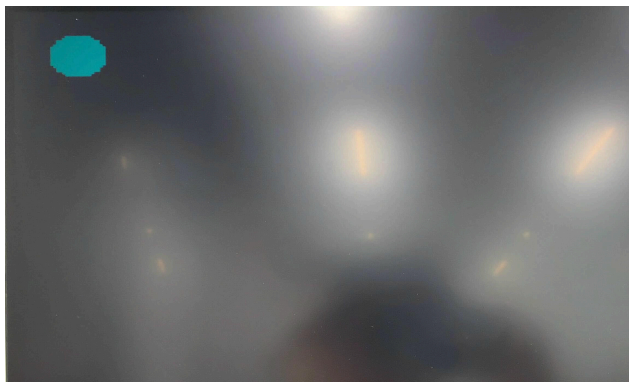


図 3: 3-2 の表示

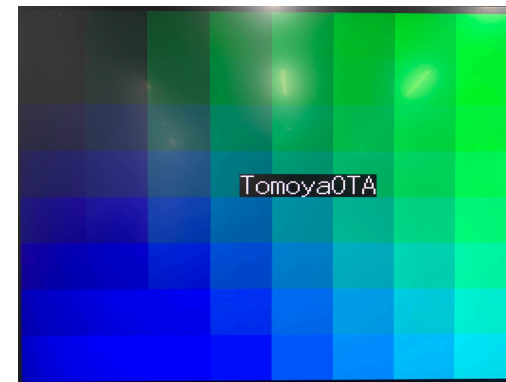


図 4: 3-3 の表示

## 課題 3 (VGA による画面表示)

- ALU は課題 2 ほど使っていないが、メモリと PLL を使用した

表 5: 課題 3 の使用リソース

	3-1	3-2	3-3
Logic usage	118	214	220
ALUT usage	190	360	359
Dedicated logic reg	95	213	247
IO pins	96	96	96
Total block memory bits	589824	589824	610304
Total PLLs	1	1	1
Max fanout	157	275	325
Total fanout	3207	4298	4732
Ave fanout	5.83	5.13	5.33

表 6: 課題 3 の最大動作周波数(@1.1V, 0°C)

	3-1	3-2	3-3
pll divclk	125.42 MHz	75.57 MHz	108.33 MHz
oVGA_HS	209.12 MHz	188.93 MHz	193.39 MHz

## 課題 4 (音声入出力)

- 4-1: 入力された音声をそのまま出力
- 4-2: 周波数を設定して正弦波を出力

## 課題 4 (音声入出力)

- メモリ使用量が3同様大きく, 4-1 では fanout も他と比べ大きい

表 7: 課題 4 の使用リソース

	4-1	4-2
Logic usage	224	162
ALUT usage	314	262
Dedicated logic reg	253	178
IO pins	106	106
Total block memory bits	2097152	196608
Max fanout	432	125
Total fanout	9723	2007
Ave fanout	9.36	2.95

表 8: 課題 4 の最大動作周波数(@1.1V, 0°C)

	4-1	4-2
AUD_BCLK	185.36 MHz	160.82 MHz
CLOCK_500	197.71 MHz	186.15 MHz
CLOCK_50	274.65 MHz	287.19 MHz
I2C	343.41 MHz	418.76 MHz



## 課題 5

コンセプト: 格子ボルツマン法による流体シミュレーション

- 簡単な離散的モデルで, 流体中の粒子の運動を逐次計算する
- 結果を VGA で(できればリアルタイムに)表示する

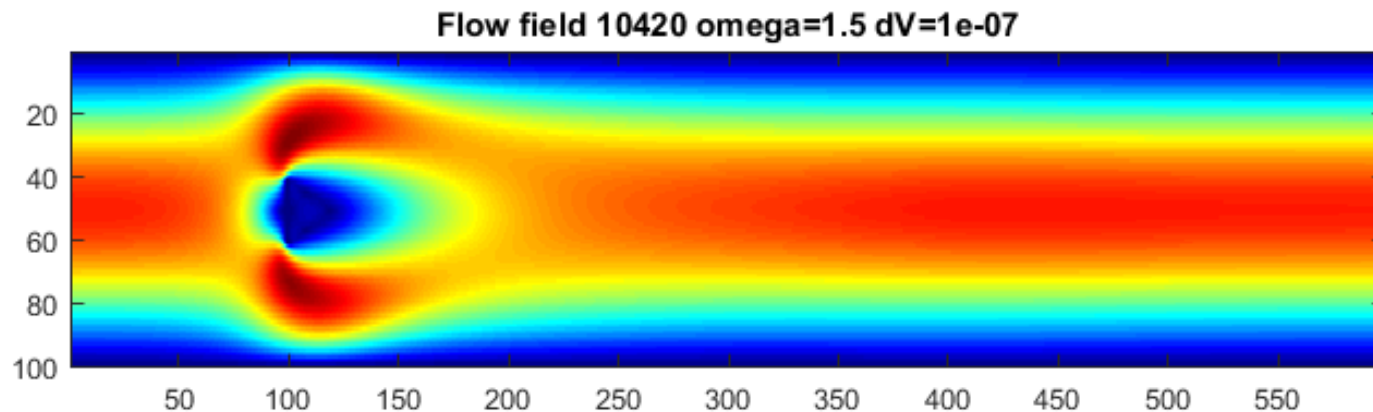


図 5: 完成イメージ([1] より引用)

## 課題 5(格子ボルツマン法のアルゴリズムの概要)

- 流体は粒子の集合で, 2次元の離散的な格子上に分布するモデル
- 次の時刻に隣接する格子に移動するか静止(物体に衝突した粒子は  $180^\circ$  反射する)
- このモデルのもとで, 各時刻での粒子の密度や速度分布を更新

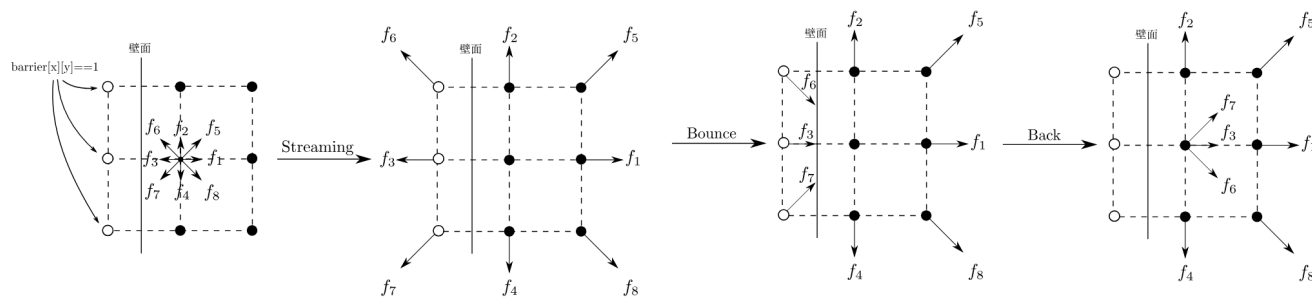


図 6: 格子ボルツマン法の流れ([2] より引用)

## 課題 5(アーキテクチャ(?))案

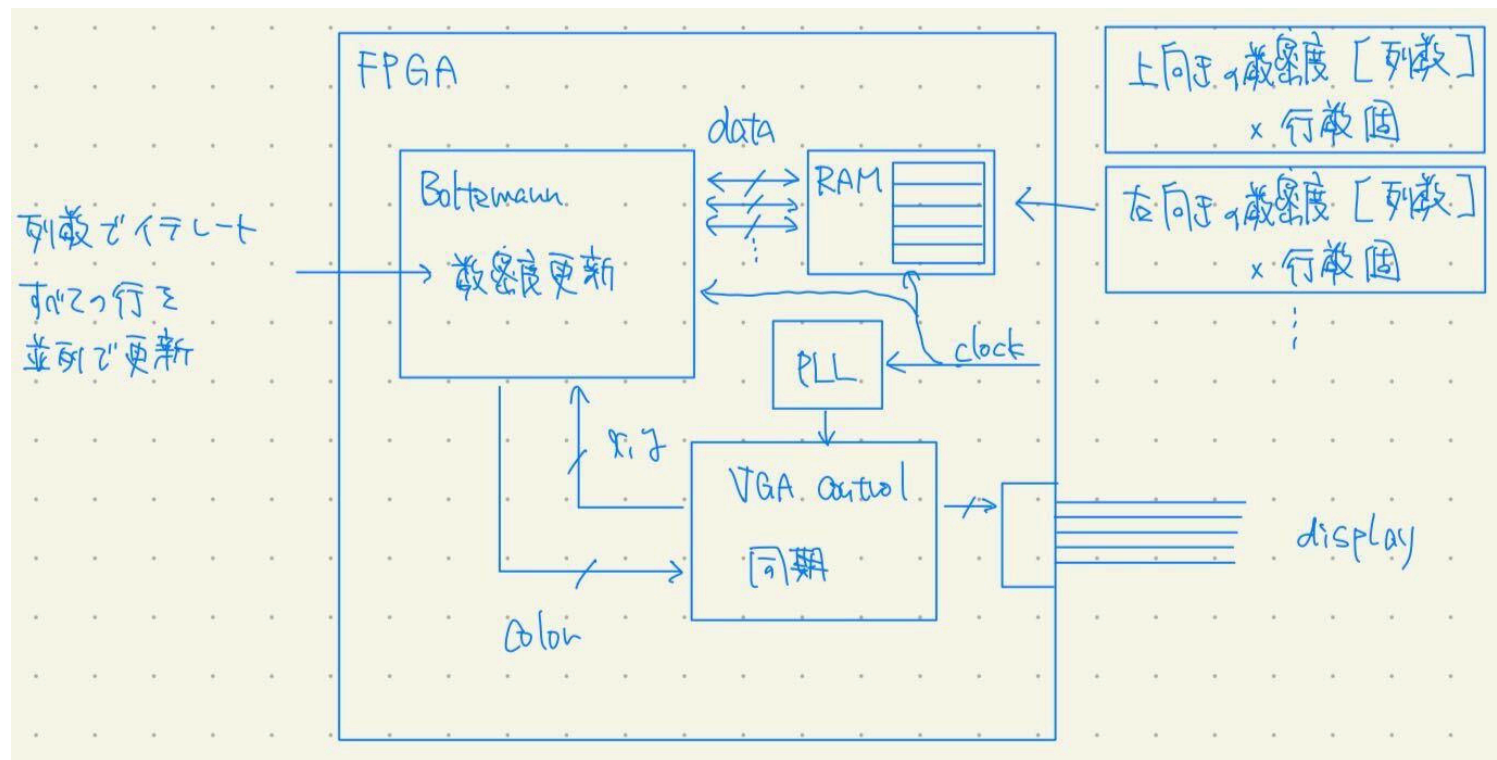


図 9: アーキテクチャ

## 課題 5(FPGA での実装の利点<sup>1</sup>)

- 各格子点で独立に流入してくる粒子の情報を処理すればよいいため、並列計算に向いている
- 複雑な制御フローがない
- メモリアクセスが規則的かつ局所的である

---

<sup>1</sup>実際に FPGA での高速化が報告されている [3]

## 参考文献

- [1] Cornell University, 「Lattice Boltzmann Cornell ece5760」. 参照: 2025 年 10 月 8 日. [Online]. 入手先: [https://people.ece.cornell.edu/land/courses/ece5760/DE1\\_SOC/HPS\\_peripherals/Lattice\\_Boltzmann\\_index.html](https://people.ece.cornell.edu/land/courses/ece5760/DE1_SOC/HPS_peripherals/Lattice_Boltzmann_index.html)
- [2] 「格子ボルツマン法による流体シミュレーション (Python)」. 参照: 2025 年 10 月 8 日. [Online]. 入手先: [https://salad-bowl-of-knowledge.github.io/hp/physics/2018/01/30/lattice\\_boltzmann.html](https://salad-bowl-of-knowledge.github.io/hp/physics/2018/01/30/lattice_boltzmann.html)
- [3] K. Sano, O. Mencer, と W. Luk, 「FPGA-based Acceleration of the Lattice Boltzmann Method」, 2007. [Online]. 入手先: <http://comparch.doc.ic.ac.uk/publications/files/ken07parcfd.pdf>