The American University in Cairo
School of Sciences and Engineering

A Simple Simulated-Annealing Cell Placement Tool

By:

Mario Ghaly

Fekry Yasser

Roaa Bahaa

Under the supervision of professor :

Mohamed Shaalan

Abstract

This project implements The Simple Simulated-Annealing Cell Placement Tool which minimizes the total wire length using the half-perimeter wire length (HPWL) algorithm to estimate the wire length of any net. By adjusting parameters such as the cooling rate and the acceptance probability of bad moves, we aim to analyze the algorithm's performance across various parameter values.

1. Introduction to Simulated Annealing

Simulated Annealing (SA) is an optimization technique and placement algorithm that is inspired by the thermal annealing process. It was developed by Kirkpatrick et al. (1983) and Cerny (1985) [1] . SA is particularly useful for finding near-optimal solutions to optimization problems. The algorithm begins with an initial random solution and iteratively explores neighboring solutions, accepting worse solutions or bad moves with a certain probability determined by the "temperature" parameter. This temperature decreases over time according to a cooling rate, allowing the algorithm to gradually focus on more promising areas of the search space. SA has been successfully applied to a variety of problems, including the Traveling Salesman Problem (TSP) , and layout optimization, which is the objective of this project [2]. Figure 1 shows the pseudo-code of the SA algorithm that is retrieved from Dr. Mohamed Shallan slides for Digital Design II course [3].

```
Create an initial random placement
T = T_init   // Very high temp
while(T > T_final)
    Pick 2 random cells and swap them
    calculate the change in WL (ΔL) due to the swap
    if (ΔL < 0) then accept
    else reject with probability (1 - e^(-ΔL/T))
    T = schedule_temp()
```

Figure 1

With Figure two showing the expected change in the total wire length with the change in temperature [3]. At the end of the paper we will analyze whether or not this was achieved by our code.

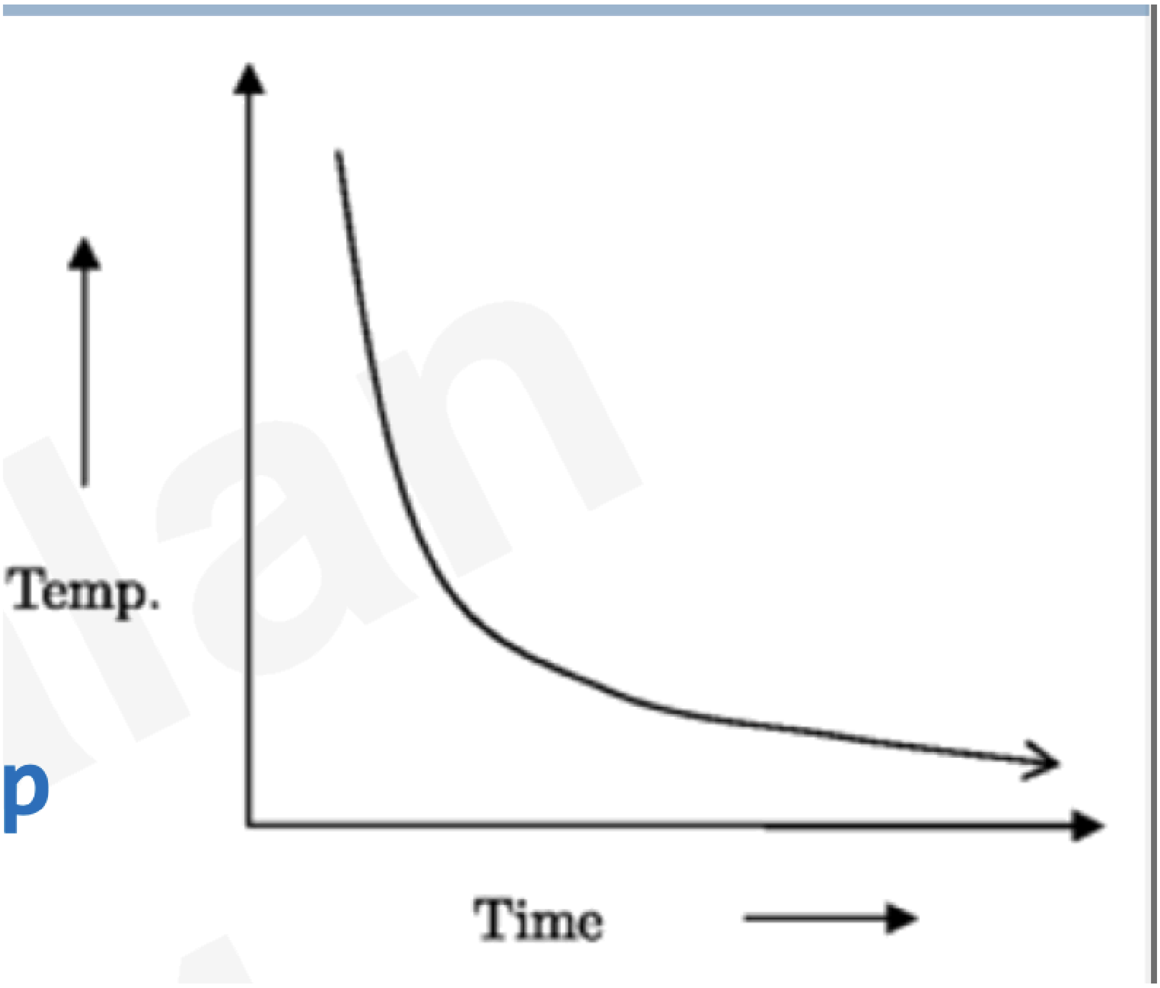

Figure 2

2. Testing

Unit testing was performed throughout the implementation phase to test each function on its own.

Also, comprehensive testing was performed using the test cases (d0,d1,d2,d3,T1) provided by the project description.

3. Code explanation

3.1. Netlist struct

```
struct netlist {
    int num_cells, num_nets, rows, cols;
    vector<vector<int>> Floorplan;
    vector<vector<int>> nets;
    vector<pair<int, int>> placed_cells;
    vector<vector<int>> cell_nets;
};
```

Figure 3

The netlist struct is used to represent a data structure that holds information related to the floor plan where:

- int num_cells: Represents the number of cells in the given text file.
- int num_nets: Represents the number of nets.
- int rows: Represents the number of rows in the floorplan grid.
- int cols: Represents the number of columns in the floorplan grid.
- vector<vector<int>> FloorPlan: A 2D vector that represents the floor plan grid. Each element of the vector indicates whether a cell is placed at that location. The value -1 means that this cell is empty.

- vector<vector<int>> nets: A 2D vector that holds information about the nets. Each element represents a net and contains indices of cells that are connected by this net.
- vector<pair<int, int>> placed_cells: A vector of pairs representing the coordinates of cells that have been placed on the floor plan grid.
- vector<vector<int>> cell_nets: A 2D vector that stores information about which nets each cell is connected to. Each element represents a cell and contains indices of nets that the cell is part of.

3.2. netlist parse_netlist(string filepath)

A simple function to parse the given .txt file and fill in the information of the net list struct. It takes the file path as an argument and returns the netlist resulted from parsing the file.

3.3. void random_placement(netlist &mynet) function

This function is responsible for randomly placing cells initially on the floor plan grid. It initializes a vector of cell indices, shuffles them randomly, and assigns each index to a grid position. If the number of cells exceeds the grid size, empty positions are marked with -1.

3.4. void BinaryGrid(const netlist &mynet)

A simple function that takes a netlist and prints it in the binary format.

3.5. int TWL(netlist &mynet)

The TWL (Total Wire Length) function calculates the total wire length of a given netlist placement on a floorplan grid. It iterates through each net, determines the bounding rectangle (min_x, min_y, max_x, max_y) that encloses all cells connected by the net, and computes the wire length as the sum of the differences between the maximum and minimum coordinates in both dimensions (x and y). In other words, it uses HPWL to calculate the TWL. The function returns the total wire length for the entire netlist.

3.6. int partial_TWL(const netlist &oldnet, const netlist &newnet, int c1, int c2)

The partial_TWL function calculates the change in total wire length (TWL) when two cells are swapped in a netlist. It first computes the wire length for each net involving the swapped cells in both the original and the new netlists. The function identifies the nets affected by the swap, calculates the bounding box coordinates for the cells in these nets, and sums the differences. It returns the difference between the new and old wire lengths, helping to determine the impact of the swap on the overall wire length.

3.7 void SA(netlist &mynet, double cooling_rate)

The SA function uses Simulated Annealing to optimize the placement of cells in a netlist, aiming to minimize the total wire length (TWL). The process starts by calculating the initial TWL and setting the initial and final temperatures. The main loop continues until the temperature drops below the final threshold. Within this loop, the algorithm iteratively selects two cells to swap and computes the change in TWL (delta_L). If the new configuration reduces the TWL or meets a probability-based acceptance rule(That is set to be “$1 - e^{-\Delta L}$"), the swap is accepted; otherwise, it is rejected. The temperature is gradually reduced by a cooling rate. After completing the annealing process, the function outputs the final placement and TWL.

4. Results

4.1. Output

The following screenshots shows the output of the 5 test files provided by the project description (d0,d1,d2,d3,t1). The output shows the placement before the SA and after SA and the time taken to compute this output.

4.1.1. D0 output

```
d0.txt

---Before SA---
00010111
10010000
10000000
10000000
 Placement:
 12   1   9  --   0  --  --  --
 --  15  18  --  14  13  23  16
 --   3  20  17   8  19  22  10
 --   7   6  11   5   4   2  21
Total wire length = 96

---After SA---
10000011
00000001
00000001
10000011
 Placement:
 --  13  14  19   3   5  --  --
  2   7   6  23   8  10  18  --
 20  22  15  12   9  17   4  --
 --   1  16  11  21   0  --  --
Total wire length = 36
Time taken by function: 0.323883 seconds
```

Figure 4

4.1.2. D1 output

```
d1.txt

---Before SA---
00000010
00000100
01000000
00000000
00010000
 Placement:
 12   1   9  24   0  27  --  31
 29  15  18  28  32  --  23  33
 26  --  20  17   8  19  22  34
 30   7   6  11   5   4   2  21
 14  35  10  --  25   3  13  16
Total wire length = 193

---After SA---
10000001
10000000
00000000
00000000
10000000
 Placement:
 --   1  23  32   9   6   7  --
 --   0  30   3  15  25  29   8
 13  26  22  33  21  34  14  10
 20  28   4  19  35  24  31  17
 --  11  18   2  12   5  27  16
Total wire length = 64
Time taken by function: 0.492102 seconds
```

Figure 5

4.1.3. D2 output

```
d2.txt

---Before SA---
01010000100000000000
01000000000100000000
00001000000001000000
01001000010000010000
00001000001100010111
10000011000010000000
00000000000000000010
00000000000000100100
00000000000000100000
01000010010000000000
00000000000000000010
01001100000000000000
00000100000000010000
00000000000010000010
00000000000001010100
 Placement:
205  -- 162  -- 237 136 227 152  -- 102 193 124 244 219  71 150 246 108  94  17
175  --  64  66  65   7 222  76 138  99 256  -- 192 155  56  39  25   3 151 101
 61  68 172 167  -- 160 257  19 179 180 198  33  21  -- 197  15 120 196 239  44
 83  --  46 103  --  95 213 200 170  --  70 149 163 148 251  -- 211 189 100  42
191  52  72  53  -- 215 245 207 144 146  --  --  93 147 153  -- 116  --  --  --
 --  16  90 127 255 202  --  -- 178 221 229  10  -- 105 177  58   0  60 234  34
 50 141 242 156 134  28 176 118  48   1 210   6 187 119 243 128 232 166  -- 236
137  73 253 217 214 212  69 111  79 184  51  87 216 135  -- 259 249  --  35 182
107 235 126  31   2 254 133 233  18 201 174   8 159  82  -- 113   9  63  41 104
 59  -- 231 145  89 122  -- 142  86  --  29  30 169 154 223 112 140  78  75 220
238  54 143 195 131 130 228 139 121 109  12 161 125  57  74 190 250 171  -- 199
 40  -- 204 247  --  --  98 185 208 248 241  14  27 203 218 230  43  96  22 194
129  11 115 165  32  --  91  36 157 117 225 206  20 106  62  --  77  97 183 158
132 240 168  67 181  23 173   5  45  81 188  47  -- 209 164  55 110  49  -- 226
 37  80 252 224   4 114  84  85  26  13  88 258 186  --  92  --  24  --  38 123
Total wire length = 3788
```

Figure 6

```
---After SA---
11111110000000000111
11111100000000000111
10011100000000000000
00000000000000000000
10000000000000000010
11000000000000000001
01000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000001
01000000000000000000
11100101111000000010
 Placement:
 --  --  --  --  --  --  --  80 186   1 170 141  40  68   2  41 196  --  --  --
 --  --  --  --  --  --  51 252 181 240 191 244 152 162  25  24 250  --  --  --
 --  20 247  --  --  --   8 231  91 235 131 258  48  64 171  12 217 118  15 238
 95  76  73  96 183 206   7  32  57 219 137  45  74 237 200 105 134 100 216  81
 -- 157 194 125  26 146  60  84  50  33 148  55  56  89  11 211  53  82  -- 248
 --  --  58 198 257  18 187  71  70 138 256 124 159 192 176 142 253 245 204  --
 23  -- 122 120 179 123 178 135  61   0 147  43  78 243 164  37 229  47 143  16
 30 184 154 103 172 180 236 199 107  83 241 113  93 130  49 188 215 249 155 251
165 111   3 255 119 185  59  27  90  22 153 228 156  44  75 224 133  46 212 189
239 116 201 173 174  35 110 209  77 233 163 182 175 160 102  38  63 104 115 169
 36 114 150 128  29 167 225 108 109 140 202  92 132 195  67 230 158 190  17 112
 14 214 223 127 242 144   6  31 232  86  98 227 193 168 207  69 197  19   4 218
220 221 205 129 126 121  87 226  13  85  54  94  97  79  34  66  62 136  65  --
  5  -- 208 145 222  52 177  21  42  28 203  99  72 101  10 149 106 117 210 161
 --  --  -- 254 259  -- 213  --  --  --  --  88 246   9 234 151 139 166  --  39
Total wire length = 980
Time taken by function: 7.37829 seconds
```

Figure 7

4.1.4. D3 output

```
PS C:\Users\EGYPT\Simulated-Anealing> ./a
d3.txt

---Before SA---
0111111011101100111001111
0110011100000000001101100
0111000010110010100100001
1000000001110001110010111
1010111101101110011010100
1000000001010110011100000
0010111100010011010000101
0000001100010010001100001
1000001001000000111111011
1000110011110001001011000
1011010100010000000000101
0001100111000110111010110
0100011101100000011000100
1001001111000010001010010
0111001010111000101000100
 Placement:
205  --  --  --  --  --  -- 152  --  --  -- 124  --  --  71 150  --  --  --  17 175  --  --  --  --
  7  --  -- 138  99  --  --  -- 155  56  39  25   3 151 101  61  68 172  --  -- 160  --  -- 179 180
198  --  --  -- 197  15 120 196  --  44  --  --  46 103  --  95  -- 200 170  --  70 149 163 148  --
 -- 211 189 100  42 191  52  72  53  --  --  -- 207 144 146  --  --  -- 147 153  -- 116  --  --  --
 --  16  -- 127  --  --  --  -- 178  --  --  10  --  --  --  58   0  --  --  34  -- 141  -- 156 134
 -- 176 118  48   1 210   6 187 119  -- 128  -- 166  --  -- 137  73  --  --  -- 212  69 111  79 184
 51  87  -- 135  --  --  --  --  35 182 107  -- 126  31  --  -- 133  --  18 201 174   8  --  82  --
113   9  63  41 104  59  --  -- 145  89 122  -- 142  86  --  29  30 169  --  -- 112 140  78  75  --
 --  54 143 195 131 130  -- 139 121  --  12 161 125  57  74 190  --  --  --  --  --  -- 204  --  --
 --  98 185 208  --  --  14  27  --  --  --  --  96  22 194  --  11 115  --  32  --  --  36 157 117
 -- 206  --  --  62  --  77  -- 183 158 132  -- 168  67 181  23 173   5  45  81 188  47  -- 209  --
 55 110  49  --  --  37  80  --  --  -- 114  84  85  --  --  88  --  --  --  92  --  24  --  -- 123
192  --  50 129  64  --  --  -- 159  --  -- 154 199  21  26  66  83  --  --  19 193 102  --  90 164
 -- 136 108  --  43 171  --  --  --  --  60 165  40  20  --  94 109 162  -- 167  --  33  93  -- 105
 76  --  --  -- 186  28  --   4  --  13  --  --  -- 106  91  38  --   2  -- 203 202 177  --  97  65
Total wire length = 3691
```

Figure 8

```
Total wire length = 3691

---After SA---
1111111111110111111111111
1111111000000000101111111
1101000000000000000011111
1110000000000000000011111
1100000000000000000000111
1110000000000000000000111
1110000000000000000000111
1100000000000000000000111
1100000000000000000000111
1100000000000000000000111
1100000000000000000001111
1100000000000000000001111
1111010000000001101101111
1111111111111111111111111
 Placement:
  --  --  --  --  --  --  --  --  --  --  -- 149  --  --  --  --  --  --  --  --  --  --  --  --  --
  --  --  --  --  --  --  --   3 153 209  59 133 119 165 207  60  --   9  --  --  --  --  --  --  --
  --  --  98  --  58  81 134 113 154 158 148 156 138  51 118 150 120 107 125   2  --  --  --  --  --
  --  --  -- 144 109  54  99  76 130 198  31 155  18 112 212  13 146 160 159 127  --  --  --  --  --
  --  -- 169  14 185 178 200 168 101 152  27  30 188  11  29  19 124 123 122   0  24 196  --  --  --
  --  --  -- 131 179   5  55 166  77 202 197  57 173  40  79 186 126 140 142  52  23  42  --  --  --
  --  --  -- 192 203  63 106 145  20 129  17 137  97 201 151 117  33 114  37 132  65  67  --  --  --
  --  --  22 102 171  21 105  34  85 205   4 199   6  89 139 194  90  41 116 195 100 172  --  --  --
  --  --  46  28  49  92 108 174 190 206 210 104  78 135  43 111  70 115  16  45  82  53  --  --  --
  --  -- 161  62  50 184 163  88  15 162  66  39  96  73  91 176  38  32 110 187 170 157  --  --  --
  --  -- 181  80  93 128  84 183 191 204 193 141  25  44 103  26  36  83  56  86 211  --  --  --  --
  --  --  71 208  10  87 182 175  75  95  35 121   8 143  61  64  12   7  94 147 189  --  --  --  --
  --  --  --  --  69  -- 180  74  47 136 167   1 164  68  72  --  --  48  --  -- 177  --  --  --  --
  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
Total wire length = 867
Time taken by function: 12.7755 seconds
```

Figure 9

4.1.5. T1 output

```
PS C:\Users\EGYPT\Simulated-Anealing> ./a
t1.txt

---Before SA---
00000000000000000000000000
00000000000000000000000000
00000000000000000000000000
00000000000000000000000001
00000000000000000000000000
00000100001000000100000000
00000000000000000000000000
00000000000000000000000000
00000000000000000000010000
00001000000010000000000000
00000000001000000000000000
00000000010000100001000000
00000000000000000000000000
00000000000000000000000000
00000000000000000010000000
00000000000000001000000000
 Placement:
205 309 342 279 237 348 366 152 270 321 320 124 244 219 394 150 246 327 390 376 175 271 304 315 374   7
375 350 138  99 402 269 300 155  56  39  25   3 151 101  61  68 172 372 301 160 307 319 179 180 198 346
313 283 197  15 120 196 239  44 316 317  46 103 295  95 213 200 170 381  70 149 163 148 251 292 384 189
100  42 191  52  72  53 272 401 245 207 144 146 268 264 347 387 153 281 116 333 277 293 290  16 323  --
255 361 287 275 178 221 229  10 289 349 371  58   0 306 358  34 302 141 362 156 134 355 176 118  48   1
210   6 187 119 243  -- 232 166 267 236  --  73 386 352 214 212  69  --  79 184  51  87 216 135 262 259
249 400  35 182 107 351 126  31 367 254 385 233  18 201 174   8 308  82 280 113   9  63  41 104  59 288
310 145  89 122 276 142  86 274  29  30 169 325 223 112 140  78  75 220 238  54 143 195 131 130 228 139
121 341  12 161 125  57  74 190 250 356 273 312 345 299 204 247 291 343  98 185 208  -- 241  14  27 369
218 230 329  96  -- 194 368  11 115 336  32 260  --  36 157 117 225 206 331 363  62 282  77 373 183 158
132 240 168  67 181  23 173   5 379  81  --  47 263 209 334  55 110  49 278 226 396  80 353 224 357 114
 84  85 314 359  88 258 354 284 383  --  24 286 332 123  -- 296 380 129  64  -- 335 257 159 261 231 154
199  21  26  66 377 298 399  19 193 102 294 397 164 311 391 108 340 392 382 338 365 285 324  60 165  40
393 227  94 109 162 266 389 337  33  93 305 395  76 235 217 252 186 378 360   4 234  13 330 370 242 106
 91  38 339 398 303 203 202 177 344  97  65 222  17  83  28  45  50 265  --  92 211 133 253 147 128 167
328 136  43  20  71 105  37  90   2 322 318 215 256 111  22 137  -- 364 192 127 248 188 171 326 388 297
Total wire length = 6848
```

Figure 10

```
---After SA---
11000000000000000000000011
11000000000000000000000000
11000000000000000000000000
10000000000000000000000000
00000000000000000000000000
00000000000000000000000000
00000000000000000000000000
00000000000000000000000000
00000000000000000000000000
00000000000000000000000000
00000000000000000000000000
00000000000000000000000000
00000000000000000000000000
00000000000000000000000001
00000000000000000000000001
00000000000000000000000011
 Placement:
 --  -- 224 353 204 175  43 377  25 371 356  87 173 199 197   2  66 134  78 351  27   8  98 205  --  --
 --  -- 393  80 171  60 169  79 385   9 150 156 209  67 321 233 248 380 367 106 379 290 280 315 165  12
 --  -- 140 128  50 363 347 116 316 115 314 369   4 374 119 263 104   5 110 243 299 292  82 161 267  31
 -- 312 271 354 325 118 241 298 382 162 252 255  74  70 229 281 158 389 189  83 274  75 304 109 269 305
381 124 333  73 184 113 335 195  52 336 137 303 190 170 322 334 147 323 213  95 219  99 372  28 230 214
 41 100 121 102 125 270 211  10 163  57  55  14 295 398 108 311 326 239 235 222  56  13 375 272  26 376
 69 188 368 143  17 392  71 285 247 167 212  22 330 226 320 231 187 262 396 289 260 130  48 391 331 288
 53  89 264 251  54 145  36 373 276 182 105 207 294 307 208 361 287 301 284  45 139 168 359 291 401  49
191 228 180 220 160  29  20 273 217 324 155   0 215 340  92 349 344  38 107  81 277 256 129  23 395  46
399  90 122  96 198 176 383 397 227 275 296 223 206 127 308 149 259  93 246 313  61 306 400 142 302 237
133 357 141 200 117  37 378 365 261 111 343 310 135  76  72 253 266 174 177 360 394 254 342 179 386  35
350  77 283  18 194   6 225  84 151  86 166 196 328 178  88 186 388 352 355 236   3  65 138  68  51  42
358  85 387 185 346 332 126 123 218 242 317 146  94 339 157 240 201 193 297   7 327 362 338 366 402 265
 19 159 202  15  30 152 384 183 234  63 131 309  64 279 136 153 258 144 112 172  44 293 278 238  62  --
 91 192 268  33  21 216 244 120  16 329 345 148 203  97  40 114 181 249   1 245 286 390 210  58 101  --
364 221  34 250 257  24  32 232 103 318 319  11 154 370  59 300  47 348  39 282 164 337 132 341  --  --
Total wire length = 1868
Time taken by function: 12.1069 seconds
```

Figure 11

4.2. Graphs

For each input file, we show Cooling rate vs. TWL graph and Temperature vs. TWL graph.

4.2.1 D0 Graphs

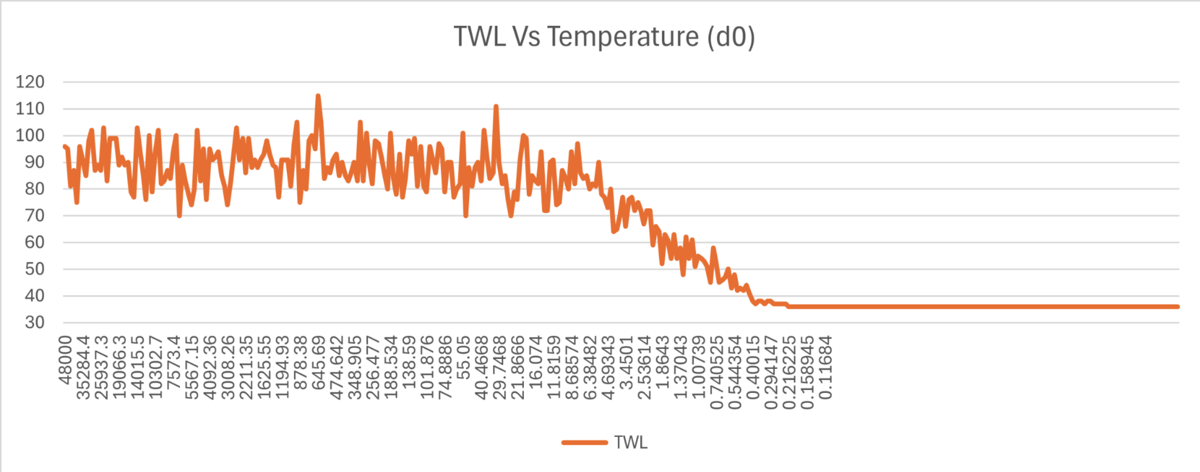

Figure 12

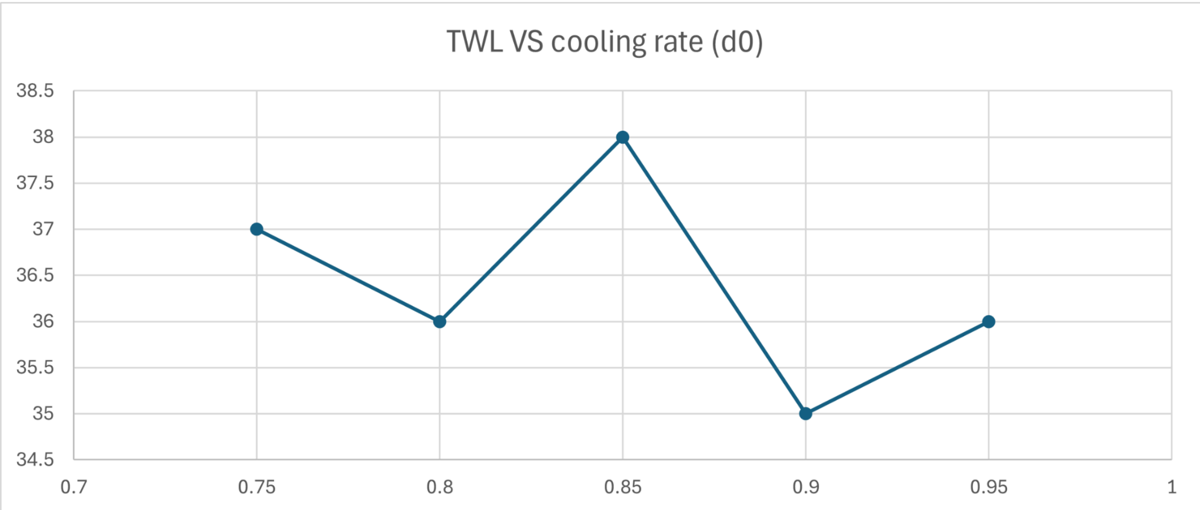

Figure 13

4.2.2 D1 Graphs

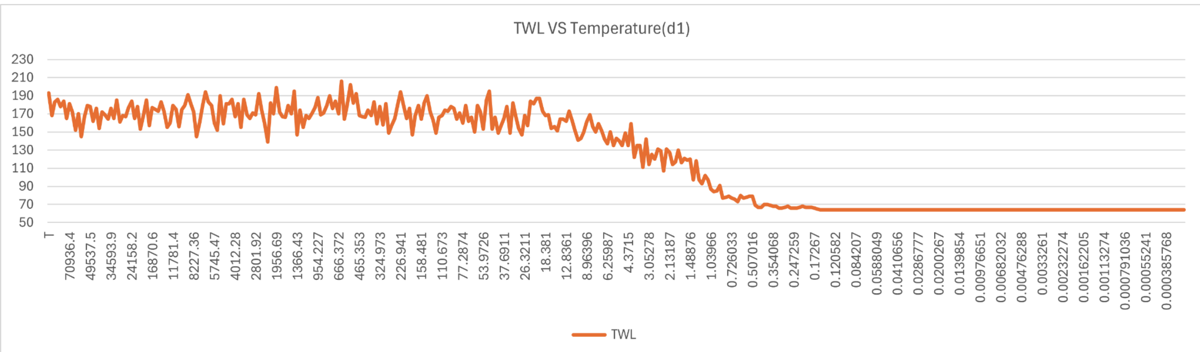

Figure 14

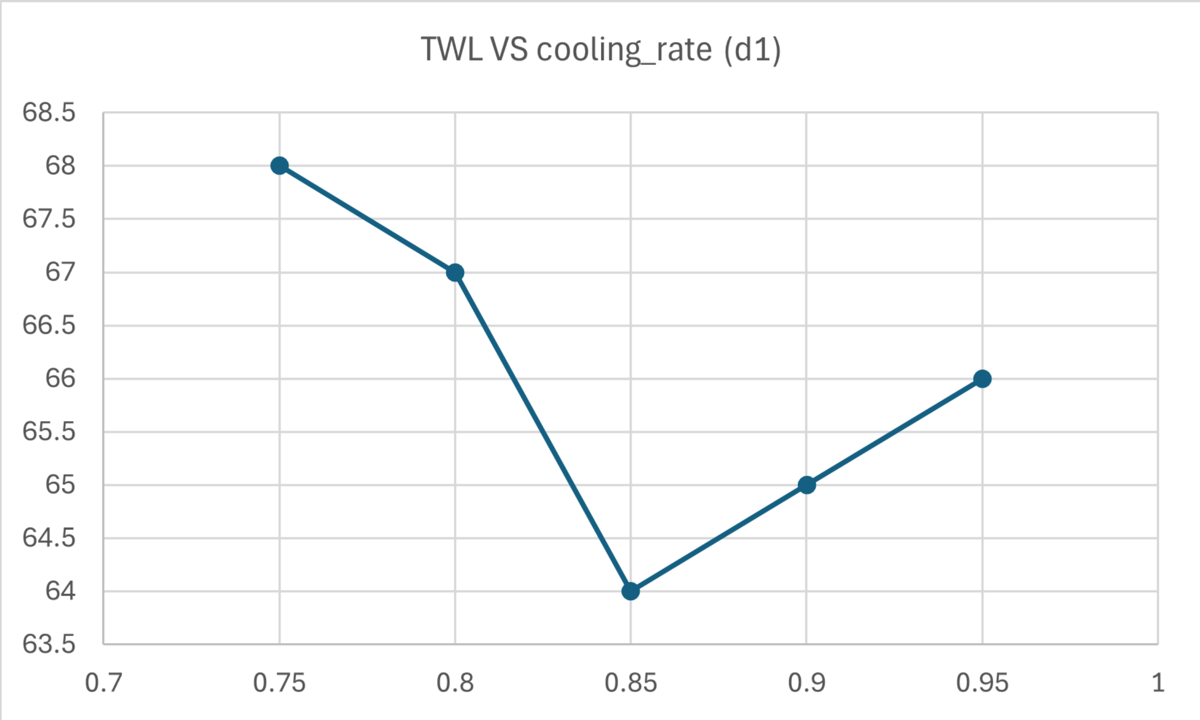

Figure 15

4.2.3 D2 Graphs

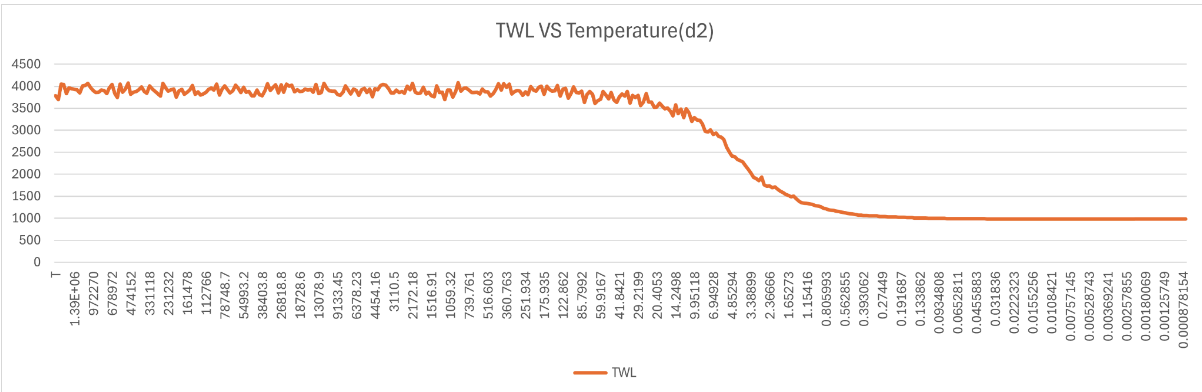

Figure 16

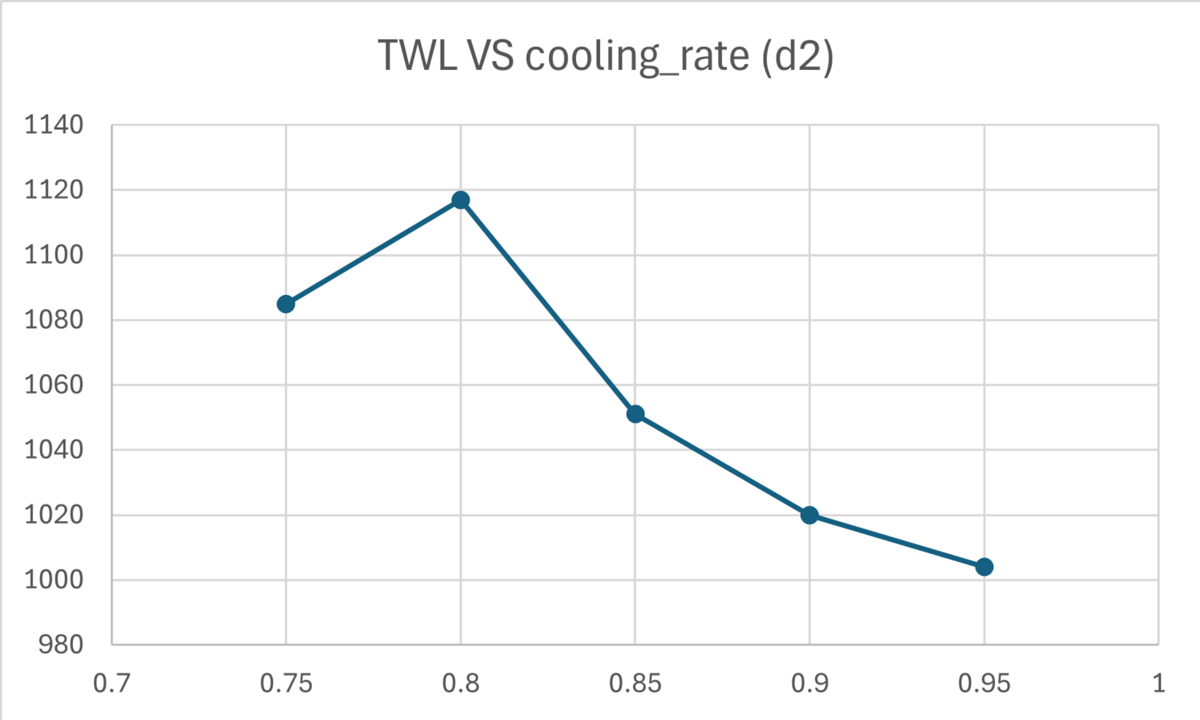

Figure 17

4.2.4 D3 Graphs

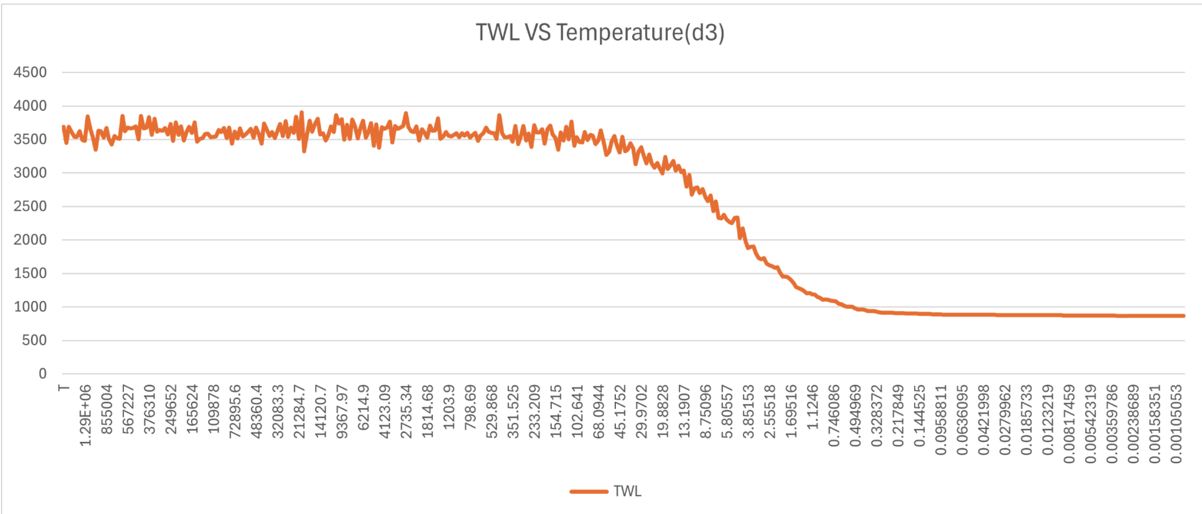

Figure 18

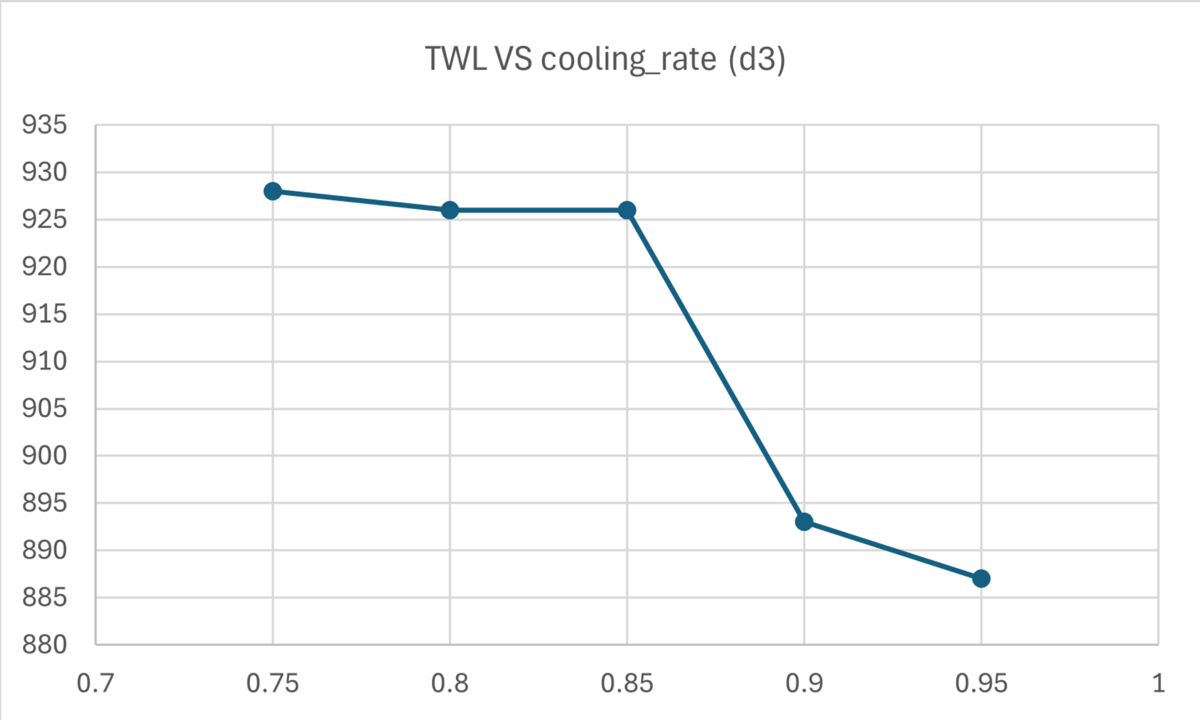

Figure 19

4.2.5 T1 Graphs

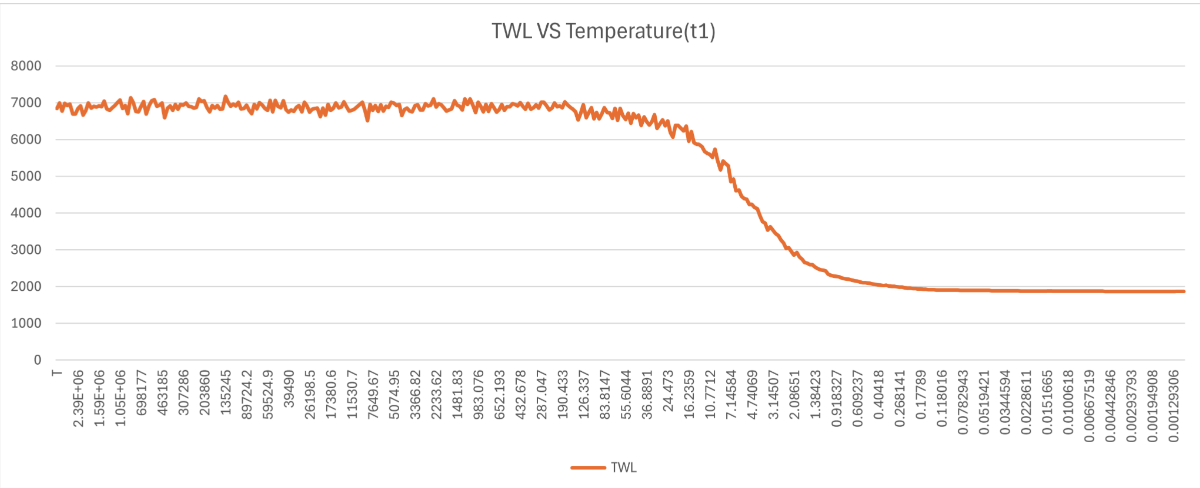

Figure 20

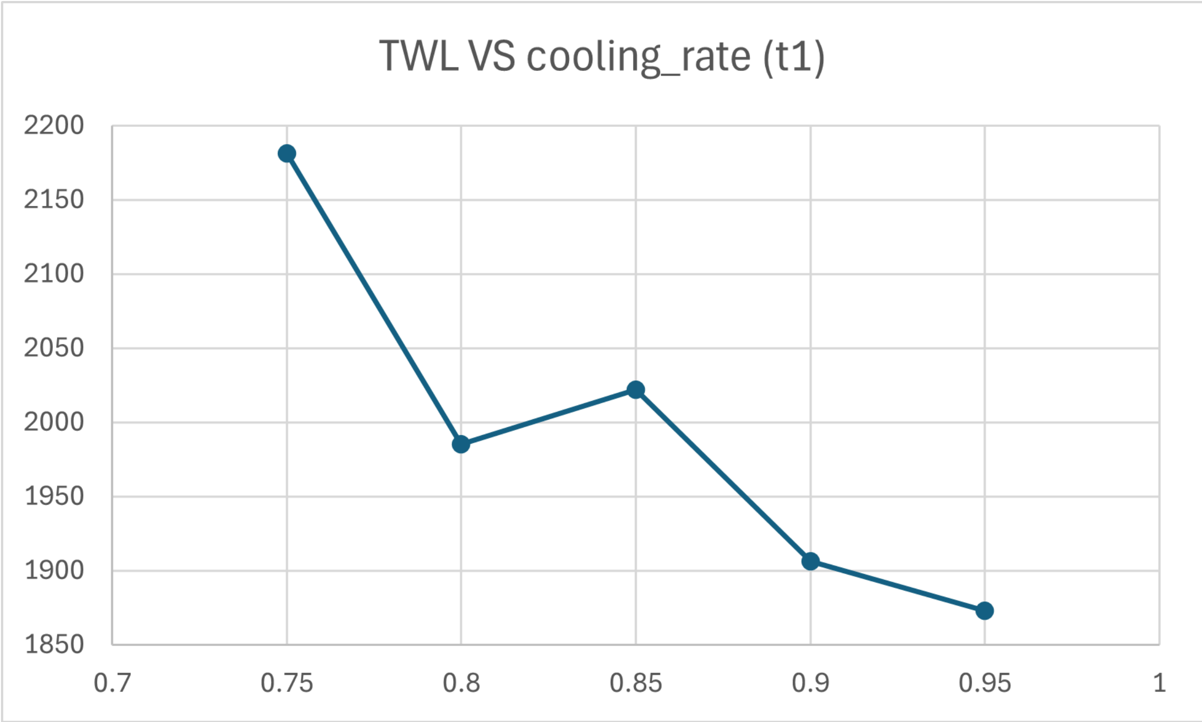

Figure 21

4.3. GIFS

4.3.1 D0 GIF

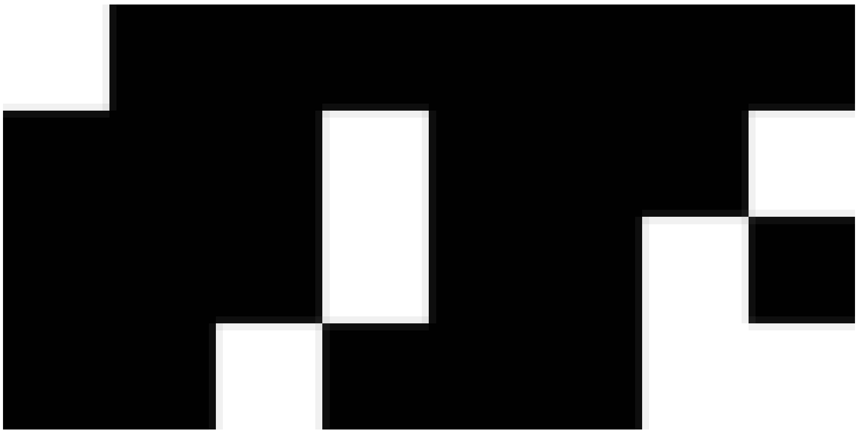

Figure 22

4.3.2 D1 GIF

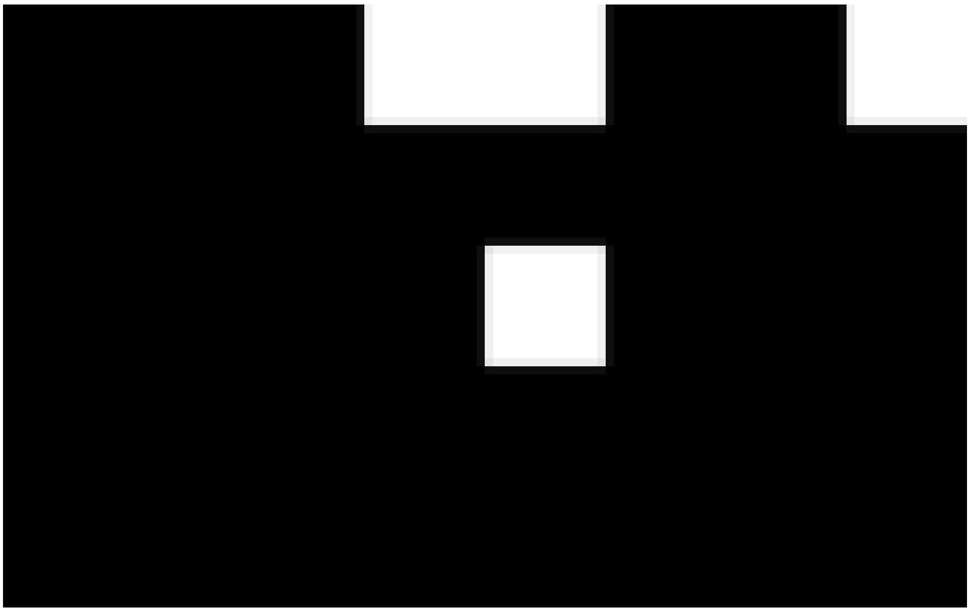

Figure 23

4.3.3 D2 GIF

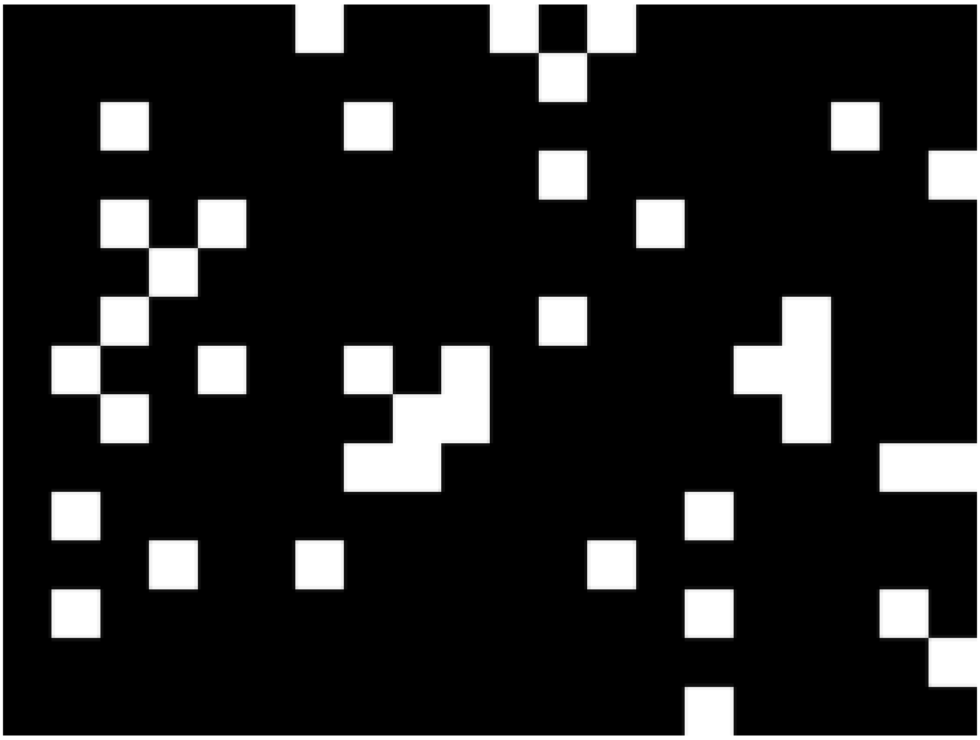

Figure 24

4.3.2 D3 GIF

Figure 25

5. Acknowledgment

We would like to express our sincere gratitude to Professor Mohamed Shaalan for his invaluable guidance and feedback throughout this project. His expertise in Digital Design II course has been instrumental in shaping our understanding of the subject matter.

6. **Conclusion**

The results, visualized through graphs and printed output, clearly show the reduction in TWL as the annealing process advances. The final placements achieved a well-distributed and optimized layout, which is clearly shown through clustering the cells in the middle of the floor plan with the empty cells outside in addition to the significantly reduced TWL, showcasing the effectiveness of the SA algorithm. Also, the lowest TWL was observed to be always the least in the range where the cooling rate is from 0.85 to 0.95. This means that usually when the cooling rate is higher, the TWL tends to be better. In conclusion, the Simulated Annealing algorithm proved to be a powerful tool for netlist cell placement optimization. Its ability to escape local minima and converge towards an optimal solution underscores its suitability for VLSI design challenges.

References

[1] “Simulated annealing,” Simulated Annealing - an overview | ScienceDirect Topics, https://www.sciencedirect.com/topics/materials-science/simulated-annealing#:~:text=Simulated%20annealing%20is%20a%20stochastic,based%20on%20the%20Metropolis%20algorithm. (accessed May 19, 2024).

[2] A. Hamdar, “Simulated annealing - solving the travelling salesman problem (TSP),” CodeProject, https://www.codeproject.com/Articles/26758/Simulated-Annealing-Solving-the-Travelling-Salesma (accessed May 19, 2024).

[3] Shaalan, Mohamed. "Digital Design II Course Slides." Lecture Slides, Department of Computer Engineering, AUC, spring, 2024.