

EECE6086 - HW 2

Max Thrun — Samir Silbak

March 24, 2014

1 Objective

The objective of this lab is to implement placement and routing algorithms capable of placing and routing the given net lists into the smallest square area possible.

2 Implementation Details

2.1 Placement

2.1.1 Placement Grid Dimensions

One of the main goals of this project is to have a final layout that is as close to a perfect square as possible. We call this the **squareness** of our layout which is the ratio of width to height ($\text{squareness} = \text{width} / \text{height}$). The closer this value is to 1 the ‘more square’ the layout is. The main issue with achieving this is that we do not know up front how tall our layout will end up since the channels between the cells are variable and the width and height of the grid directly effect them.

In order to figure out what width and height we should make our grid we need to come up with some heuristic that can estimate the squareness of a layout given only the placement grid dimensions, number of cell, and number of nets.

$$\text{squareness} = f(\text{num_cells}, \text{num_nets}, \text{grid}_w, \text{grid}_h)$$

Figure 1: Squareness Function

Once we have an equation that can estimate the squareness we can plug in different grid dimensions and figure out one that gets us the best squareness. In order to figure out the formula that estimates our squareness we manually specified various grid dimensions for each benchmark and recorded the resulting squareness. This data can be found in the Appendix section at the end of the report. We then used the program Eureqa to evolve an equation that fit our data. After running Eureqa for 15 minutes we had an equation that fit our data with an $R^2 = 0.9965$.

$$\text{squareness} = 0.341 + \frac{2.41}{\text{grid}_h} + \text{grid}_h 0.62^{\text{grid}_w} + \frac{0.303 \text{num_cells}}{\text{grid}_h^2} - 0.000162 \text{num_nets} - 0.303 \text{num_cells} 0.62^{\text{grid}_w}$$

Figure 2: Squareness Equation

To calculate the placement grid size we start the grid as a perfect square by taking the square root of the number of cells. We then decrease the height and increase the width and check the squareness of this size. We continue doing this and keeping track of which size results in the best squareness until the height is 0.

The table below shows the resulting squareness of the original method (just making a perfectly square grid) versus using the equation. We can see that by using the equation we achieved a much more square layout for all but benchmark 9. This could be a result of not having enough data points to fit a better curve with.

Benchmark	Without Equation			With Equation			Percent Improvement
	Grid Width	Grid Height	Layout Squareness	Grid Width	Grid Height	Layout Squareness	
1	7	7	0.7245	9	5	1.0789	19.66 %
2	10	9	0.7365	12	8	0.9845	24.80 %
3	14	13	0.6767	18	10	1.0914	23.18 %
4	19	19	0.7029	28	13	1.2579	3.92 %
5	27	27	0.6125	45	16	1.2060	18.16 %
6	32	32	0.6060	56	18	1.1735	22.05 %
7	23	22	0.6452	39	13	0.9823	33.71 %
8	30	30	0.6306	50	18	1.3123	5.70 %
9	29	28	0.6343	45	18	1.4323	-6.67 %
10	25	24	0.6715	36	17	1.2772	5.13 %

Table 1: Improvement from using squareness estimate equation

2.1.2 Force Directed Placement

We use a force directed placement algorithm in order to reduce the overall complexity. Our algorithm follows the one in the book exactly with the exception of what happens when the target cell is locked. In the original algorithm you search for a vacant spot to place the seed cell if the target cell is locked, we expand that to also include unlocked spots. If we choose an unlocked spot then that cell we are replacing becomes the new seed cell. We found through experimentation that this achieves less complex placements. The gray cells are cells with no connections.

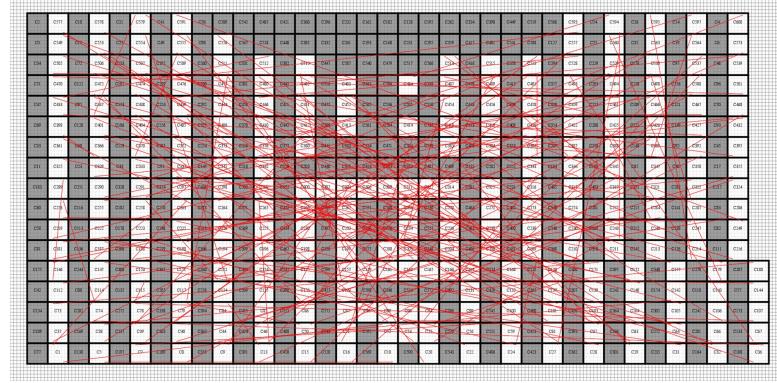


Figure 3: Original

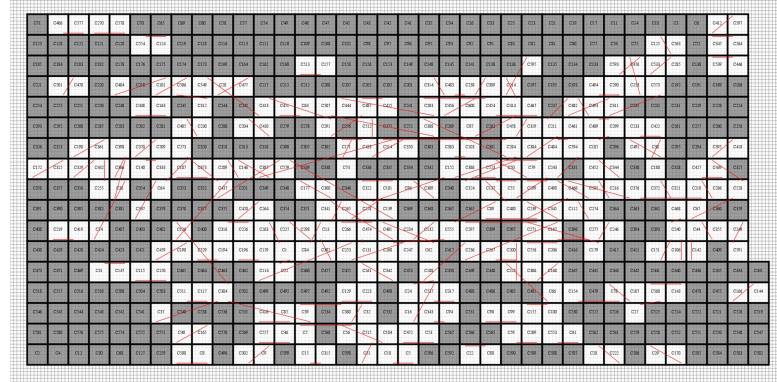


Figure 4: Force Directed

In order to figure out the best abort limit we recorded the layout area for each benchmark for various limits. We found the abort limit that achieved the minimum area and used that minimum area to normalize the areas of the others as a percent increase. We found that an abort limit of 1 gives us the lowest average percent increase in area across all the benchmarks.

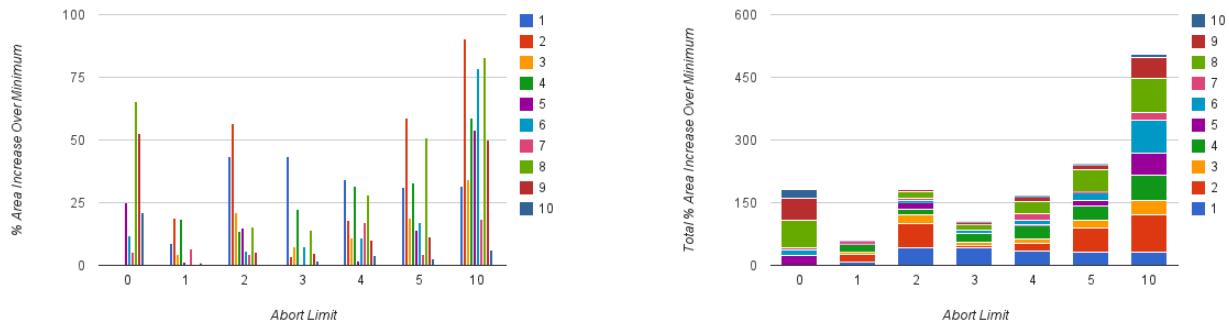


Figure 5: Percent increase in layout area versus abort limit

2.1.3 Flipping Cells

The force directed algorithm does not take into account the positions of the terminals on each cell, it assumes all connections go to the center of the cell. This results in a lot of connections that unnecessarily span vertically across a row. To fix these we simply go through each cell and try all possible flip orientations and see which one reduces the wirelength the most. This approach is not ideal as it does not take into consideration chains of cells that all may need to be flipped. An approach similar to KL where we walk down the connection tree trying flips and only actually flip as many as what improves the total wirelength would be interesting. Unfortunately we did not have time to explore this.

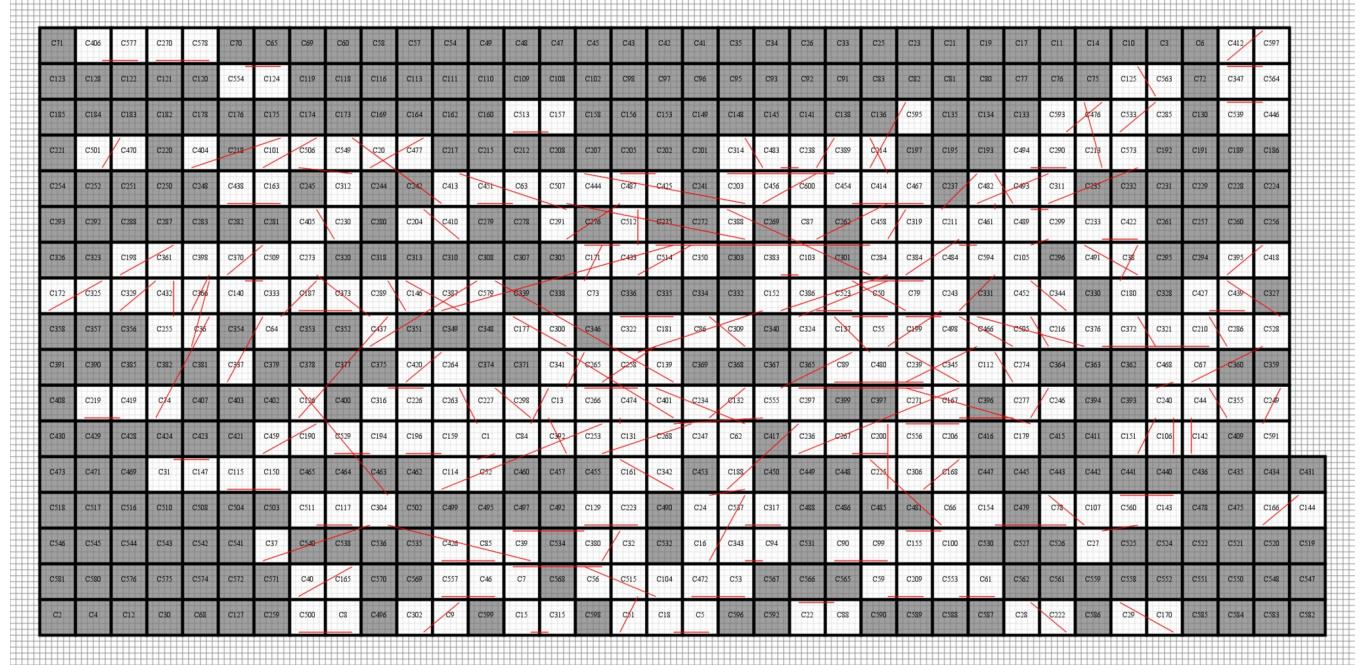


Figure 6: Original

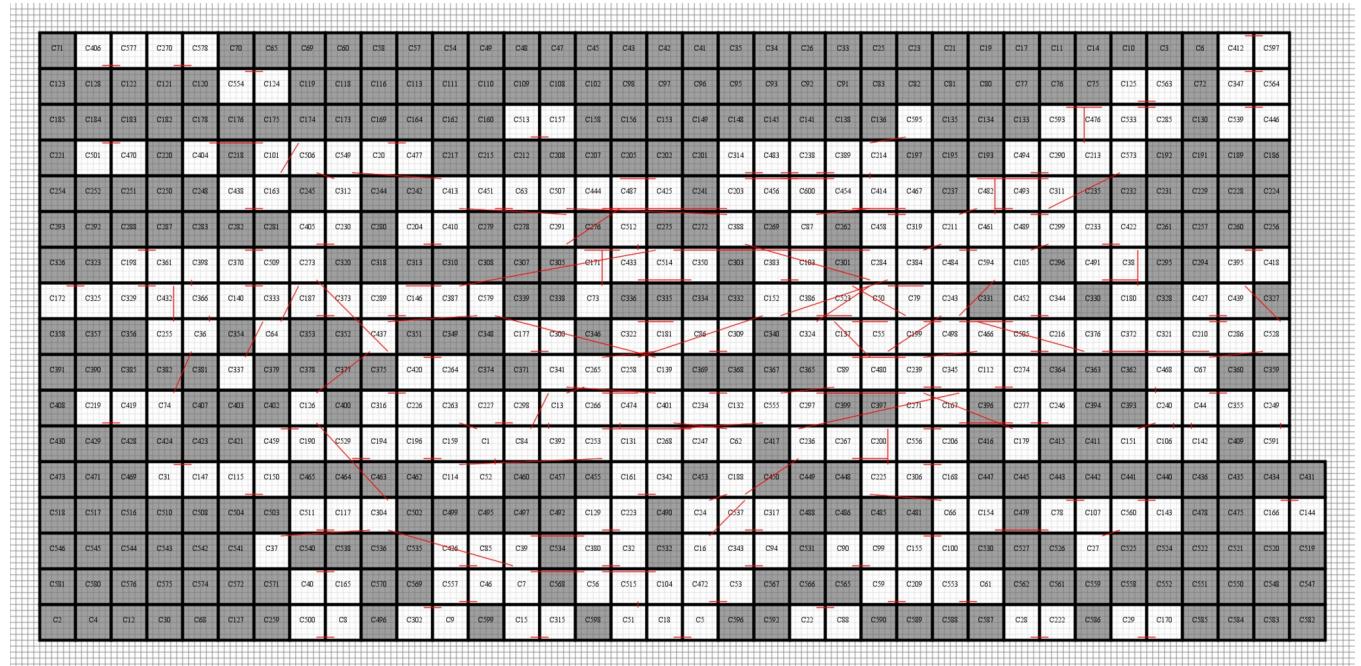


Figure 7: Flipped

2.1.4 Adding Feed-Through Cells

In order to channel route we cannot have connections that span rows. To fix this we add feed-through cells (shown in light gray) in order to transfer the nets between rows. The process of adding feed-through cells starts at the bottom row. It goes through each terminal of each cell and detects if the terminal is in the same channel as its destination terminal. If it's not, it figures out if it needs to add a feed-through to the current row (say the terminal is at the bottom of the cell and it needs to connect to somewhere above it) or if it needs to add the feed-through to the next row up (if the terminal is at the top of the cell but it connects to the top of the cell above it). If a feed-through cell is added the whole row is rescanned since we might need to add yet another feed-through cell for the feed-through we just placed.

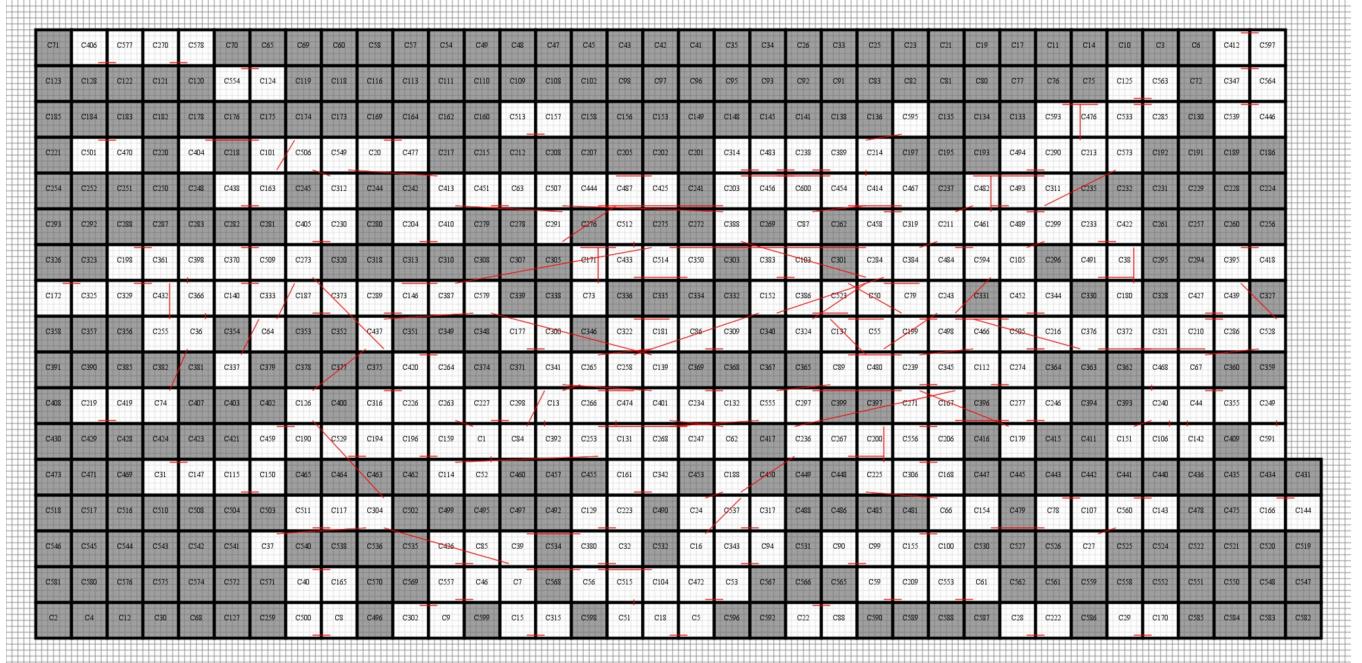


Figure 8: Original

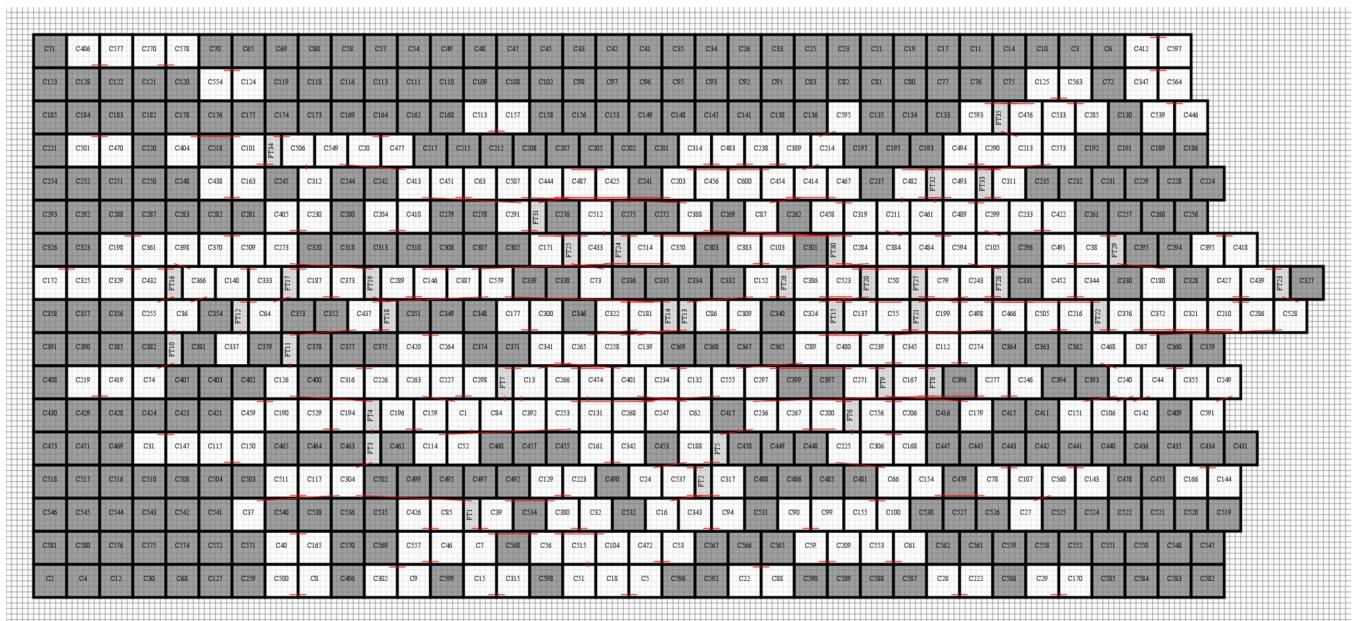


Figure 9: With feed-throughs

2.1.5 Evening Row Lengths

After adding the feed-through cells the row lengths become uneven. In order to fix this we add all unconnected cells to a queue and then go through them adding them to which ever row is currently the shortest. Additionally, we alternate placing them on either side and we also skip every other cell. The reason behind this is that since we just messed up a lot of the force directed work by removing all the unconnected cells we will have to re-force direct them within their rows. Spacing them out will allow them to more easily move around and have a higher chance of finding a vacant spot close to their target.

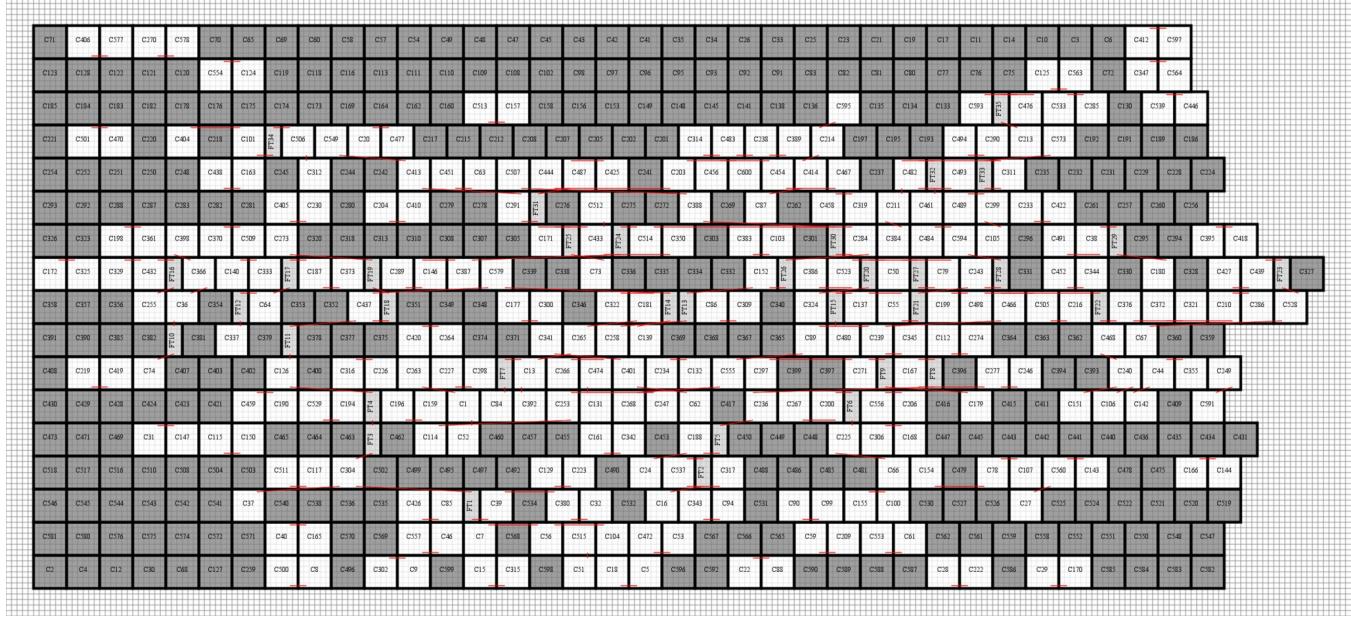


Figure 10: Original

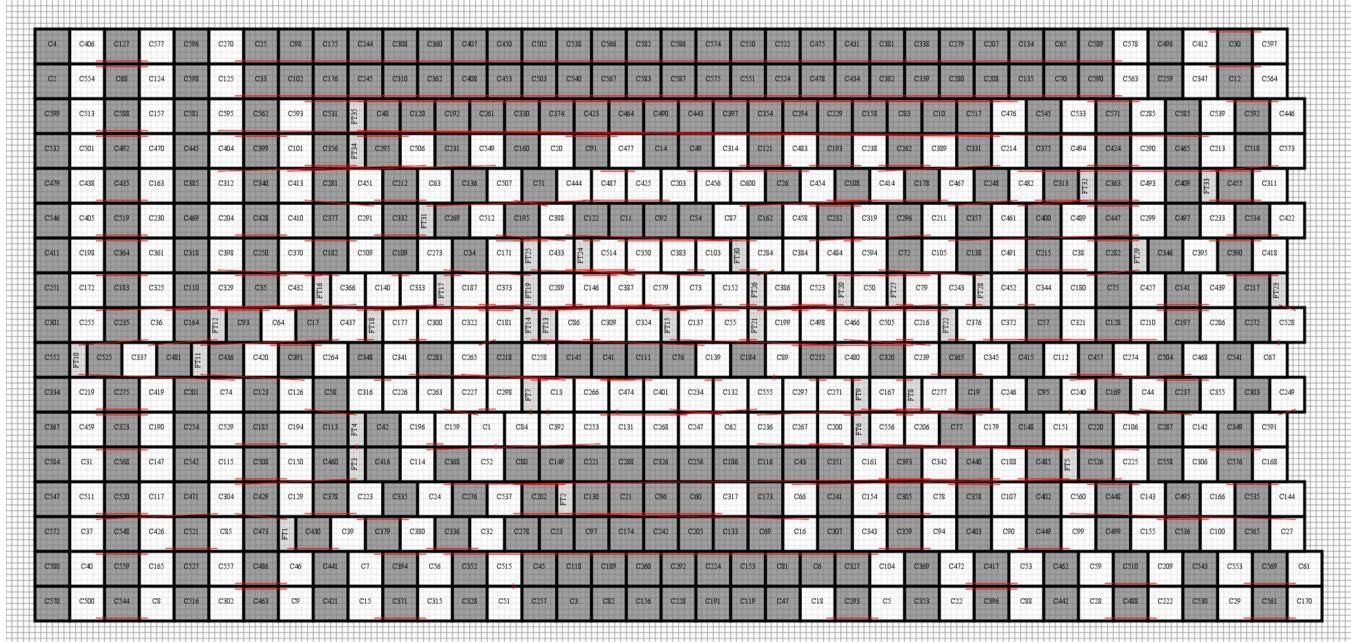


Figure 11: Even length rows

2.1.6 Pulling Cells Closer

After evening out the row lengths we are left with a lot of space between connected cells. In order to fix this we run a simplified force directed algorithm which goes through each cell, calculates its zero force location on the x axis, and then finds the closest unconnected cell to that location and swaps its position with it.

C4	C406	C127	C577	C596	C270	C25	C98	C175	C244	C303	C360	C407	C409	C502	C138	C566	C382	C398	C574	C509	C322	C475	C481	C381	C388	C279	C371	C134	C65	C389	C578	C495	C412	C30	C397							
C2	C554	C68	C134	C98	C131	C33	C102	C175	C245	C310	C360	C408	C413	C503	C140	C567	C383	C397	C575	C511	C324	C478	C482	C389	C397	C280	C308	C131	C70	C390	C563	C159	C347	C12	C364							
C99	C311	C388	C177	C381	C393	C502	C393	C511	C751	C405	C120	C192	C301	C330	C374	C413	C444	C480	C443	C397	C354	C394	C289	C318	C83	C10	C311	C476	C345	C533	C371	C385	C383	C389	C392	C446						
C321	C301	C402	C470	C445	C404	C389	C101	C158	C754	C395	C366	C331	C548	C160	C38	C91	C477	C14	C48	C314	C131	C483	C193	C238	C363	C389	C331	C14	C371	C444	C304	C386	C455	C213	C318	C373						
C479	C438	C465	C163	C383	C312	C340	C413	C381	C401	C111	C313	C63	C138	C307	C71	C444	C407	C405	C203	C46	C200	C36	C434	C414	C178	C467	C348	C402	C131	F11	C383	C481	C409	C163	C311	C397						
C546	C401	C519	C330	C469	C304	C403	C40	C377	C391	C312	C332	C369	C512	C195	C383	C322	C11	C92	C54	C37	C162	C408	C323	C319	C396	C311	C331	C461	C400	C488	C447	C399	C477	C233	C354	C412						
C411	C196	C364	C361	C318	C389	C210	C370	C312	C309	C109	C273	C344	C171	F11	C483	C754	C134	C300	C381	C103	C789	C344	C384	C164	C394	C72	C105	C138	C461	C211	C38	C382	F11	C346	C395	C390	C418					
C251	C172	C183	C335	C310	C339	C35	C432	F11	C566	C140	C333	F11	C137	C373	F11	C389	C346	C387	C379	C73	C152	F11	C383	C333	C311	C50	F11	C79	C348	C342	C452	C344	C131	C75	C407	C141	C489	C217	C3			
C381	C251	C255	C38	C164	F11	C93	C54	C17	C467	F11	C177	C300	C322	C131	F11	C381	C566	C389	C324	F11	C137	C55	F11	C199	C498	C468	C305	C218	F11	C376	C372	C57	C321	C128	C218	C197	C388	C272	C353			
C552	F11	C355	C337	C481	F11	C436	C420	C391	C344	C341	C381	C303	C381	C318	C358	C315	C454	C41	C111	C76	C139	C484	C389	C323	C480	C303	C389	C365	C345	C415	C112	C457	C324	C304	C483	C341	C367					
C334	C119	C275	C419	C381	C74	C123	C138	C5	C316	C236	C363	C227	C398	F11	C13	C366	C474	C401	C341	C132	C551	C297	C71	C152	F11	C383	C333	C31	C50	F11	C79	C348	C342	C452	C344	C131	C75	C407	C141	C489	C217	C3
C367	C439	C133	C198	C254	C329	C185	C194	C113	F11	C42	C196	C139	C1	C34	C382	C233	C131	C308	C347	C62	C236	C387	C300	F11	C556	C386	C377	C77	C179	C148	C131	C230	C387	C381	C430	C191						
C554	C39	C68	C147	C542	C113	C303	C130	C402	F11	C416	C114	C368	C52	C30	C140	C221	C328	C336	C256	C156	C116	C43	C391	C342	C440	C188	C467	F11	C136	C221	C358	C206	C376	C168								
C547	C311	C339	C117	C411	C384	C249	C129	C378	C338	C223	C333	C34	C276	C337	F11	C130	C31	C96	C36	C317	C173	C66	C341	C154	C305	C78	C338	C107	C403	C560	C448	C143	C495	C166	C331	C144						
C572	C37	C548	C426	C521	C385	C473	F11	C408	C289	C179	C388	C336	C212	C278	C23	C97	C274	C242	C325	C133	C269	C16	C301	C349	C359	C94	C403	C70	C448	C99	C499	C155	C336	C200	C563	C27						
C388	C40	C519	C165	C327	C557	C486	C46	C441	C7	C384	C56	C332	C515	C45	C118	C189	C368	C392	C234	C153	C81	C6	C327	C104	C388	C472	C417	C33	C402	C59	C310	C209	C345	C533	C369	C61						
C370	C300	C344	C8	C156	C382	C463	C9	C431	C15	C311	C315	C338	C14	C217	C3	C32	C156	C228	C191	C19	C47	C18	C393	C5	C313	C22	C386	C388	C440	C38	C483	C222	C338	C202	C39	C361	C170					

Figure 12: Original

C4	C127	C577	C596	C65	C134	C307	C279	C198	C381	C411	C415	C322	C190	C174	C566	C382	C398	C574	C509	C322	C475	C481	C381	C388	C279	C371	C134	C65	C389	C578	C495	C412	C30	C397					
C2	C68	C134	C98	C59	C70	C135	C308	C389	C382	C484	C478	C314	C511	C315	C387	C393	C57	C540	C303	C431	C408	C360	C310	C345	C103	C101	C33	C125	C563	C159	C347	C12	C364						
C99	C388	C113	C177	C381	C393	C502	C117	C10	C192	C383	C48	C354	C490	C397	F11	C474	C93	C448	C153	C1374	C174	C383	C388	C322	C303	C311	C371	C353	C383	C291	C358	C446							
C331	C336	C301	C470	C124	C444	C101	C482	F11	C48	C364	C160	C305	C133	C322	C38	C477	C14	C48	C311	C314	C483	C389	C328	C303	C193	C331	C91	C375	C484	C390	C389	C318	C373						
C479	C141	C488	C163	C413	C383	C111	C381	C401	C141	C340	C132	C136	C307	C444	C47	C403	C313	C382	C306	C407	C444	C101	C178	C7	C16	C341	C363	C131	F11	C481	C311	C453	C409						
C57	C346	C454	C130	C519	C342	C340	C377	C369	C291	C333	C34	C276	C337	C303	F11	C130	C31	C96	C36	C317	C173	C66	C345	C305	C78	C338	C107	C403	C560	C448	C143	C495	C166	C331	C144				
C411	C364	C198	C313	C389	C250	C323	C370	C309	C109	C273	C34	C171	F11	C483	F11	C134	C317	C389	C324	C134	C313	C381	C135	C484	C394	C384	C346	C105	C389	F11	C338	C481	C395	C418					
C251	C183	C172	C325	C366	C110	C29	C343	F11	C140	C333	F11	C187	C373	C379	C346	C387	C373	C152	C353	C72	C386	C323	F11	C305	C70	C343	C141	C344	C180	C35	C141	C47	C389	C217	C3				
C301	C39	C36	C17	C164	F11	C331	C564	C255	C437	F11	C177	C300	C322	C181	F11	C36	C309	C397	C137	C55	F11	C199	C486	C305	C218	F11	C304	C138	C376	C311	C372	C218	C386	C197	C272	C353			
C552	C481	F11	C305	C337	C391	C386	F11	C344	C304	C381	C411	C111	C139	C75	C365	C341	C383	C403	C183	C322	C389	C145	C345	C393	C329	C179	C334	C381	C382	C345	C330	C151	C488	C112	C374	C304	C407	C341	C367
C334	C74	C219	C49	C381	C281	C279	C123	C136	C324	C363	C227	C398	F11	C13	C366	C474	C401	C341	C132	C551	C297	C71	C19	C91	F11	C167	C346	C217	C340	C169	C337	C33	C344	C131	C377	C240	C335	C303	C348
C367	C333	C49	C198	C254	C183	C359	C194	C113	F11	C42	C196	C139	C1	C34	C382	C233	C131	C347	C347	C63	C77	C398	C367	C303	F11	C156	C356	C386	C179	C334	C338	C217	C3						
C554	C569	C21	C474	C342	C328	C115	C130	C460	F11	C416	C211	C346	C2	C30	C14	C368	C387	C326	C354	C356	C295	C42	C14	C485	C405	C323	C382	C188	F11	C321	C341	C346	C326	C348	C366	C346	C166	C344	
C547	C358	C311	C117	C411	C439	C384	C378	C129	C223	C335	C338	C303	C130	C346	F11	C2	C337	C311	C276	C30	C173	C363	C448	C302	C154	C107	C78	C34	C56	C403	C11	C495	C335	C333	C166				
C572	C321	C548	C471	C37	C35	C348	F11	C403	C389	C32	C380	C288	C278	C379	C16	C336	C374	C242	C303	C133	C289	C345	C348	C34	C448	C403	C39	C27	C336	C155	C305	C365	C389	C388	C397				
C388	C59	C40	C165	C327	C368	C357	C46	C441	C118	C189	C360	C384	C315	C353	C7	C56	C104	C392	C234	C153	C81	C6	C327	C45	C389	C417	C411	C33	C401	C110	C310	C9	C209	C345	C369	C353	C61		
C370	C344	C300	C8	C156	C463	C302	C9	C431	C371	C15	C315	C338	C314	C217	C3	C32	C156	C228	C191	C19	C47	C18	C393	C5	C313	C22	C386	C388	C440	C38	C483	C222	C338	C302	C39	C361	C170		

Figure 13: Cells pulled closer

2.1.7 Moving Feed-Through Cells

Sometimes, because it can not be known at the time of placement, the feed-through cells end up on the far side of the cell as shown in the ‘before’ picture. We make an effect to detect these and move them closer. We always want the feed-through cell to horizontally between the two terminals it is connected to. (This is done before routing begins, the example images below show the circuit post routing as it better illustrates the situation)

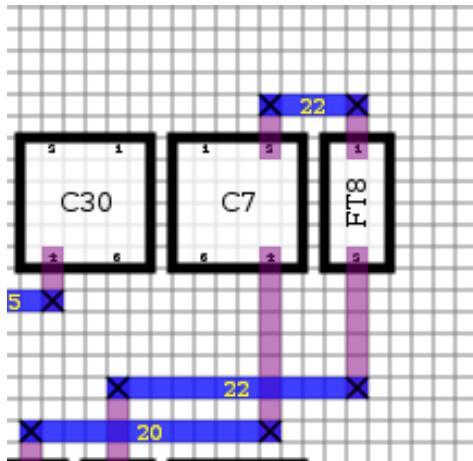


Figure 14: Before

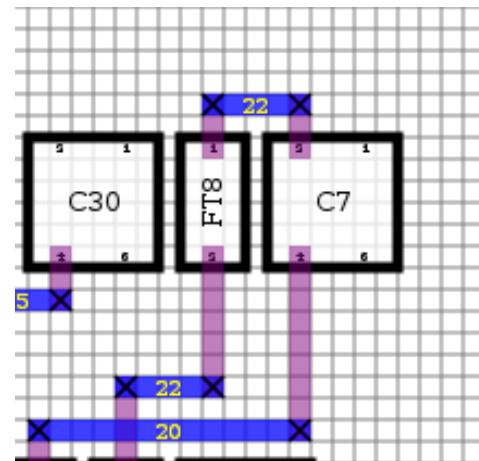


Figure 15: After

2.2 Routing

2.2.1 Assigning nets to track

Routing begins by first assigning each net to its own track. Net collisions are not considered at this point.

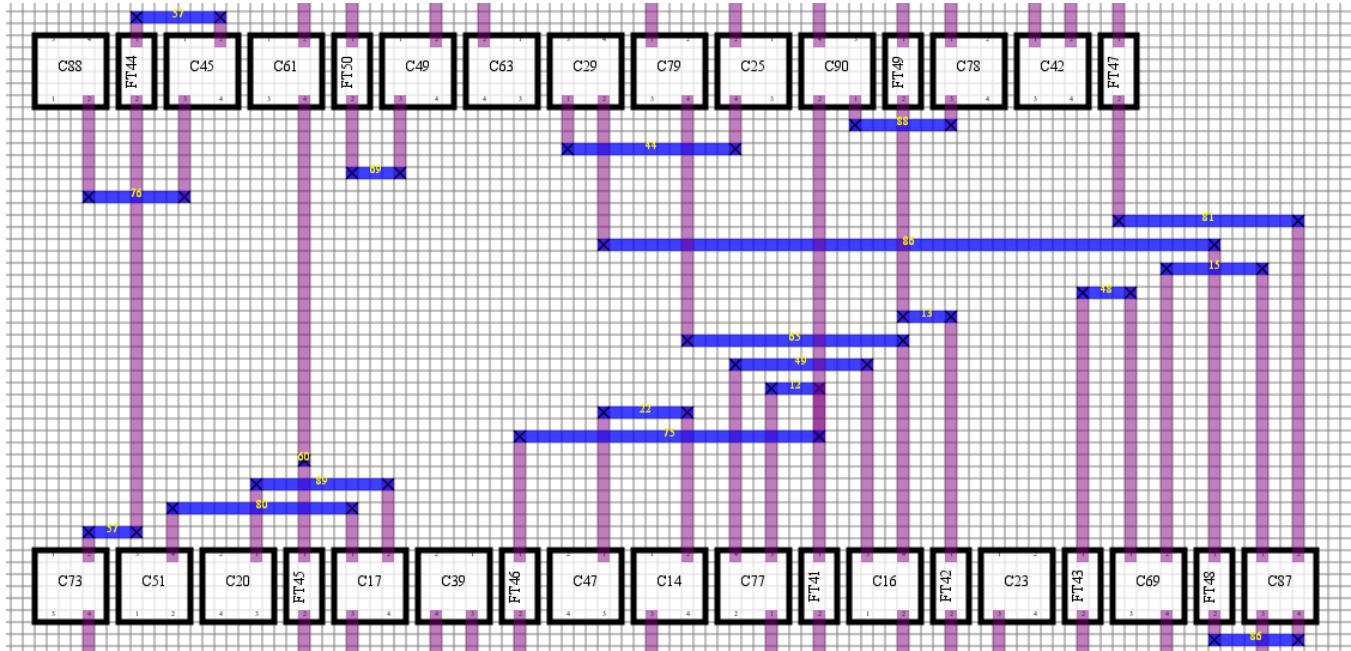


Figure 16: Each net has its own track

2.2.2 Expand the tracks

Each net is checked to see if it conflicts with another net. Some conflict configurations can be solved by adding another track to the channel (expanding the channel). The image below shows the same circuit at the previous image but with the conflict now solved and circled in red.

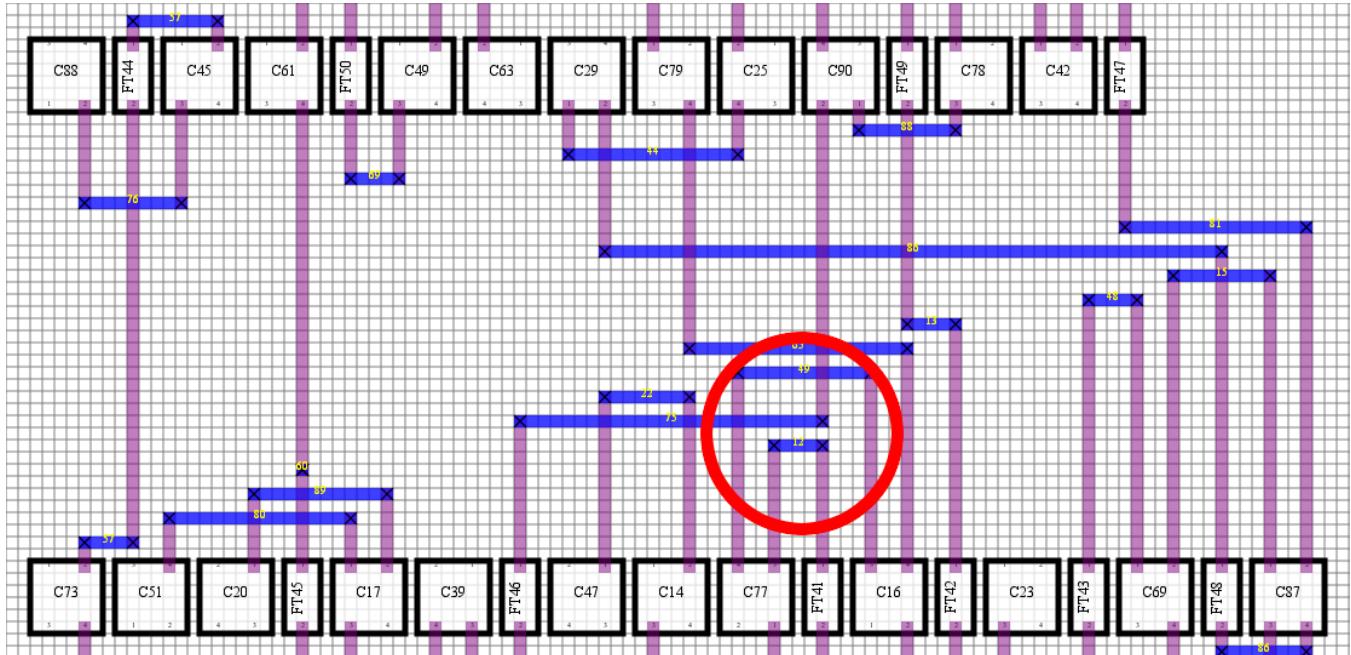


Figure 17: Conflict is solved by expanding the channel

2.2.3 Fixing overlaps

Some conflicts are true overlaps that cannot be solved no matter what net ordering you try. The only way to solve these conflicts is to move the cells in the rows over. The image on the left shows an overlap between nets 11 and 29. The only way to solve this is to move cell C36 over to the right, which is shown in the image on the right.

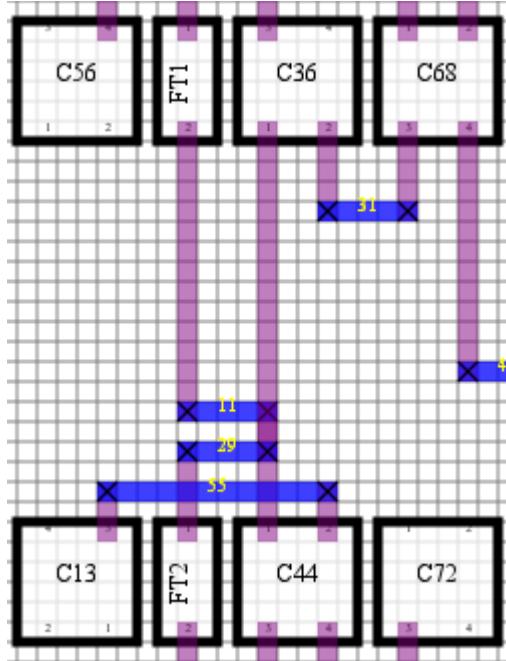


Figure 18: Overlapping nets

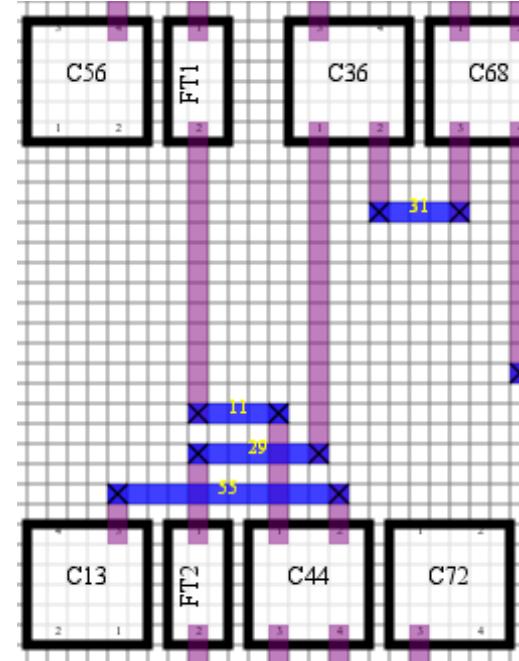


Figure 19: Conflict is solved by moving cells over

2.2.4 Finding Vertical Nets

For terminals that line up directly with their destination we can use a purely vertical net that doesn't take up a track. The image on the left shows a small horizontal segment that is currently occupying a track. We simply mark this net as purely vertical and remove it from any future track related operations.

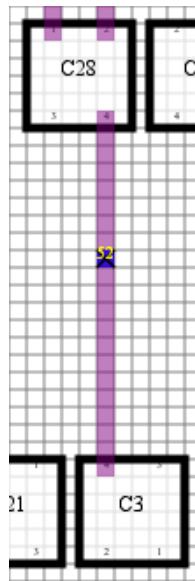


Figure 20: Vertical net taking up a track

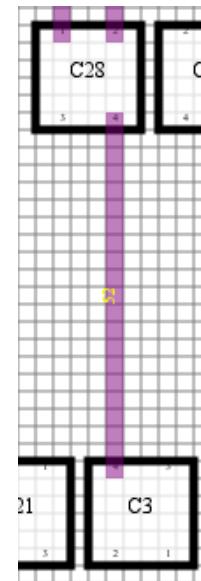


Figure 21: Unnecessary horizontal segment removed

2.2.5 Pulling Nets Up

We next want to try to shrink the channel so the first step is to pull all the nets either up or down. Nets are pulled up as far as possible without causing a conflict. We do not consider any specific ordering about which nets we pull up first although it can effect the final configuration. The image below shows the result of pulling all the nets up and then removing tracks which no longer have any nets on it.

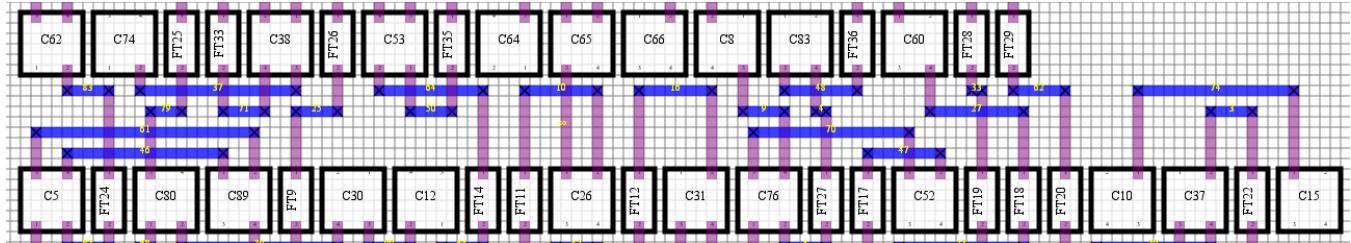


Figure 22: All nets pulled up and empty tracks removed

2.2.6 Pulling Nets In

For nets whose terminals are both on the same side of the channel we want to bring that net back to that side (pull it in to the side the terminals are on). The image below shows the result of doing that.

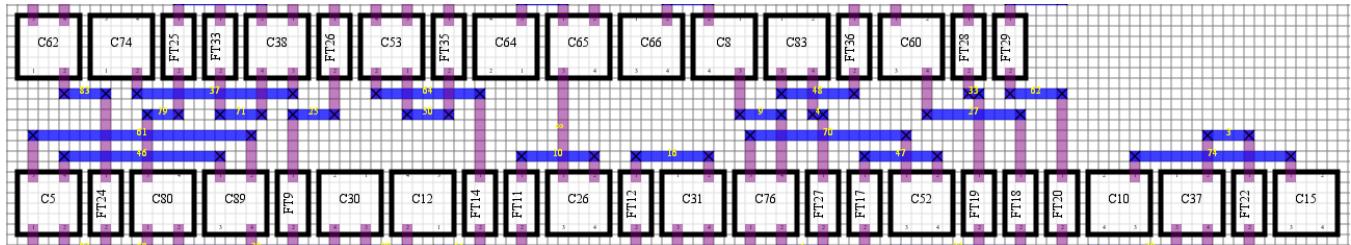


Figure 23: Nets pulled in closest to the cells they connect to

In our actual implementation we pull the cells up, then in, then down, then in again. This routine seems to be able to achieve fairly high channel density.

3 Usage

Building is accomplished via a Makefile which generates an executable names `main`. Additional build options are shown below.

```

1 $ make          # regular build
2 $ make clang   # builds using clang++ instead of g++
3 $ make debug    # enable debug printf
4 $ make clean     # removes all binaries, object files, and benchmark results
5 $ make benchmarks # runs all the benchmarks in the 'benchmark' directory
6 $ make pngs      # converts all the benchmark .svg files to .png
7 $ make jpgs      # converts all the .png files to reduced resolution .jpg files

```

The program accepts a single argument which is a file containing the netlist formatted as specified in the assignment. All log and debug output is printed to `stdout` and the output `.mag` and `.svg` files are named the same as the input file.

```

1 $ ./main benchmark/1 > benchmark/1.log
2 $ ls benchmark/1.*
3 benchmarks/1.log  benchmarks/1.mag benchmarks/1.svg

```

4 Results

The results of all the benchmark netlists are tabulated below. The wirelength is the length post routing.

Benchmark	Box X	Box Y	Box Area	# FTs	Wirelength	# Vias	Execution Time	Memory Usage
1	82	76	6232	17	1093	112	0.009873s	296 kB
2	127	129	16383	50	2547	270	0.021806s	340 kB
3	191	175	33425	112	6114	562	0.044696s	424 kB
4	317	252	79884	213	14108	1126	0.111609s	588 kB
5	486	403	195858	464	41469	2336	0.384445s	916 kB
6	602	513	308826	685	70536	3322	0.670760s	1204 kB
7	1110	1130	1254300	1738	443767	5448	1.581445s	1512 kB
8	437	333	145521	297	22907	1994	0.360278s	912 kB
9	328	229	75112	69	6097	842	0.206668s	664 kB
10	258	202	52116	35	2881	404	0.157362s	544 kB

Table 2: Result Summary

4.1 Benchmark 1 - Placement Steps

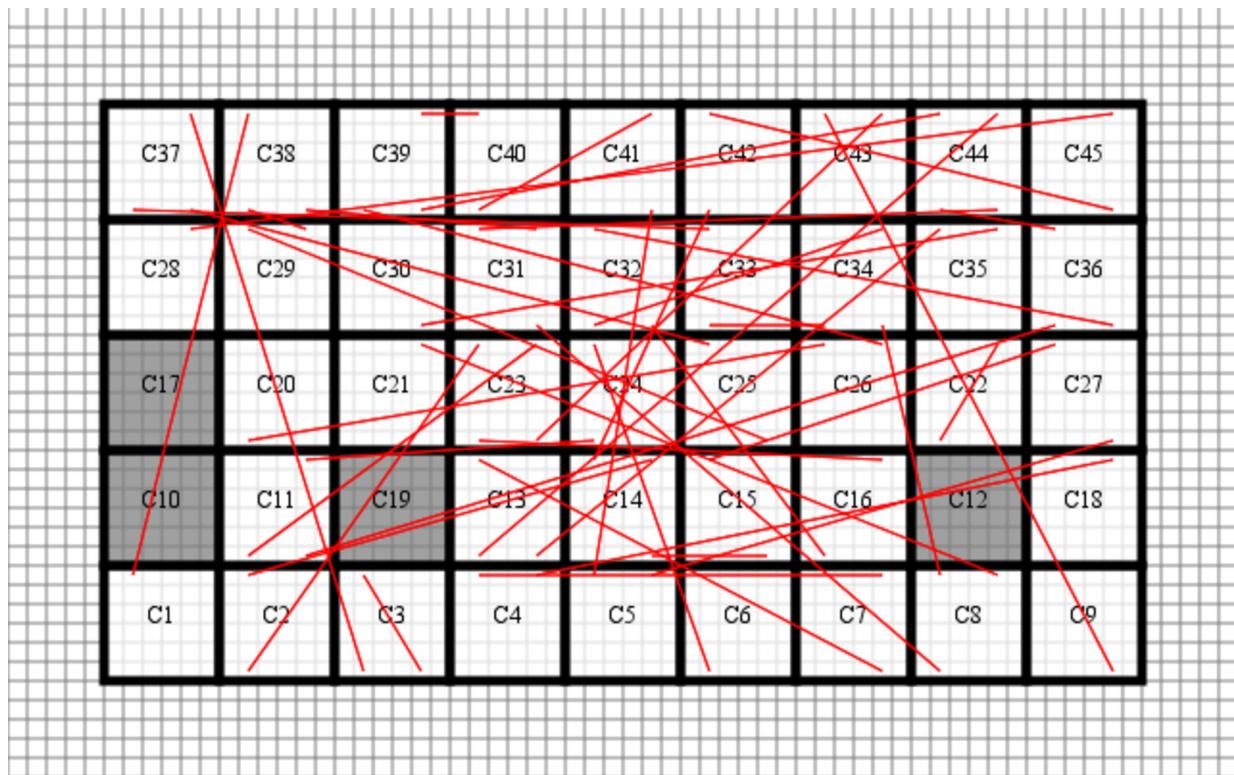


Figure 24: Benchmark 1 - Step 1/7 - Initial placement

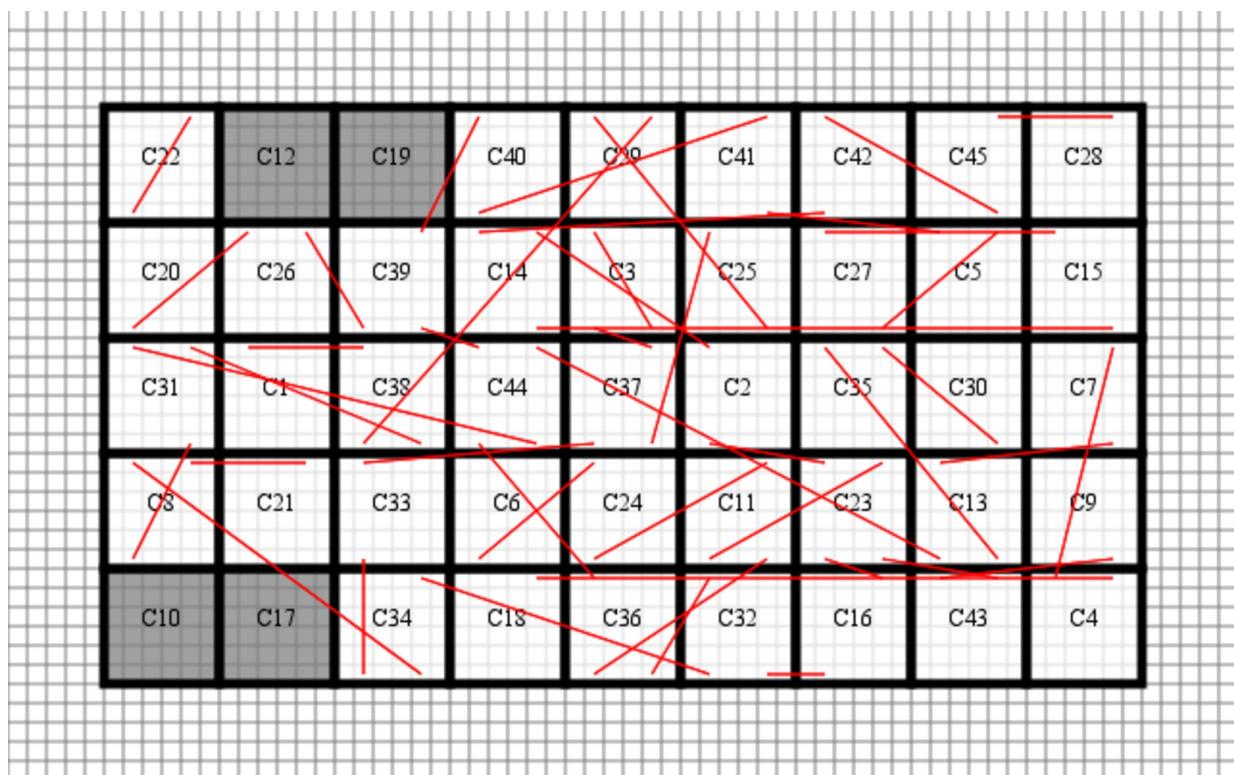


Figure 25: Benchmark 1 - Step 2/7 - Force directed placement

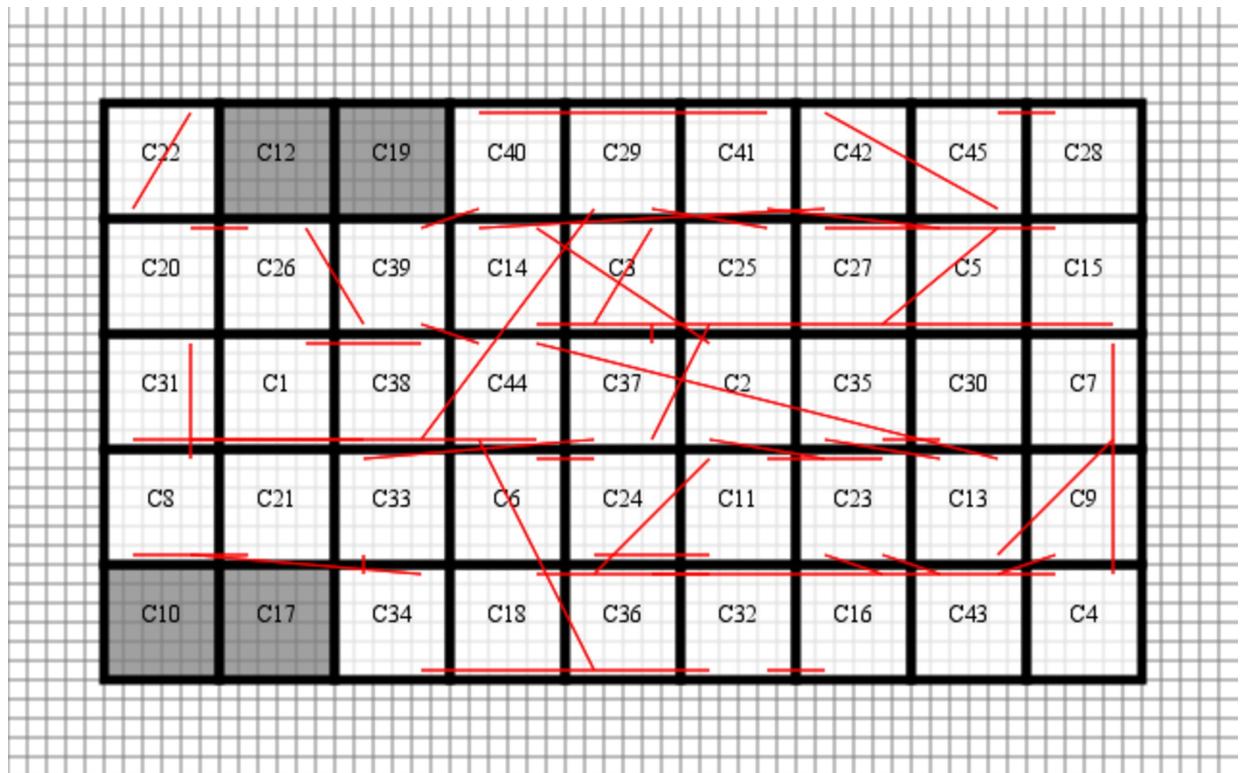


Figure 26: Benchmark 1 - Step 3/7 - Flip cells

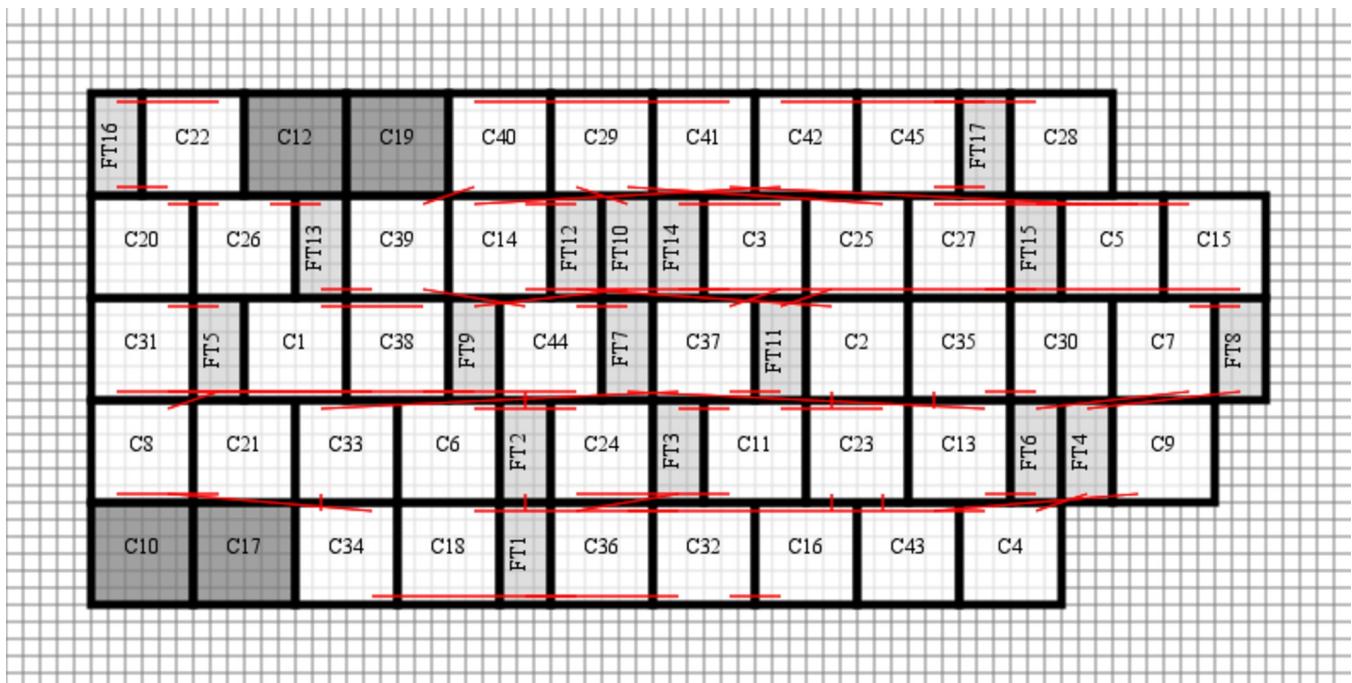


Figure 27: Benchmark 1 - Step 4/7 - Add feed-throughs

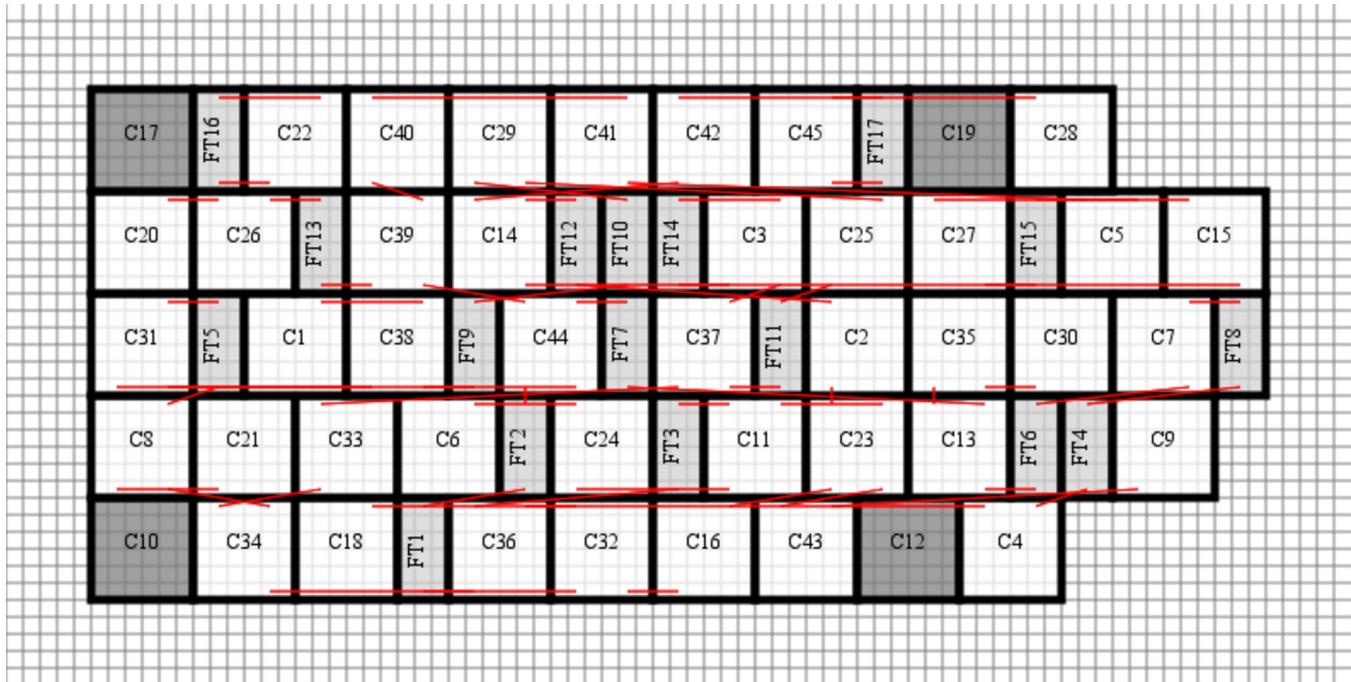


Figure 28: Benchmark 1 - Step 5/7 - Even out the row length

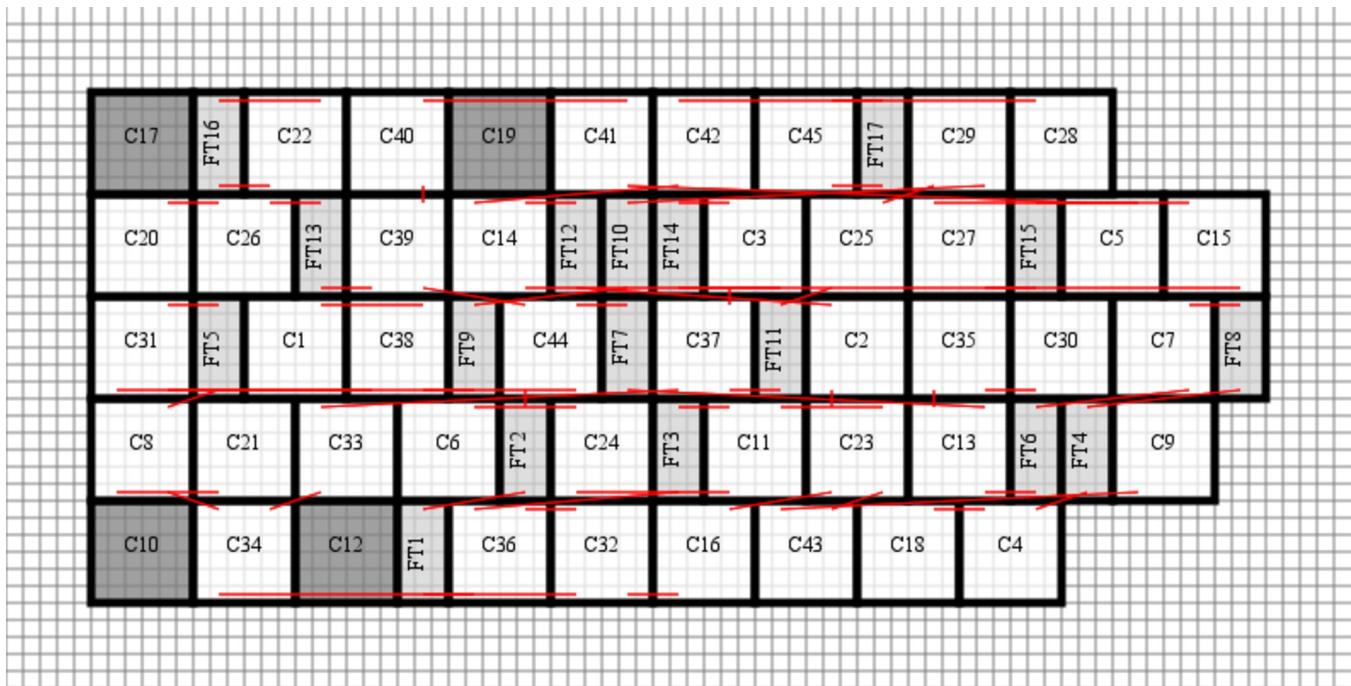


Figure 29: Benchmark 1 - Step 6/7 - Pull cells in the rows closer together

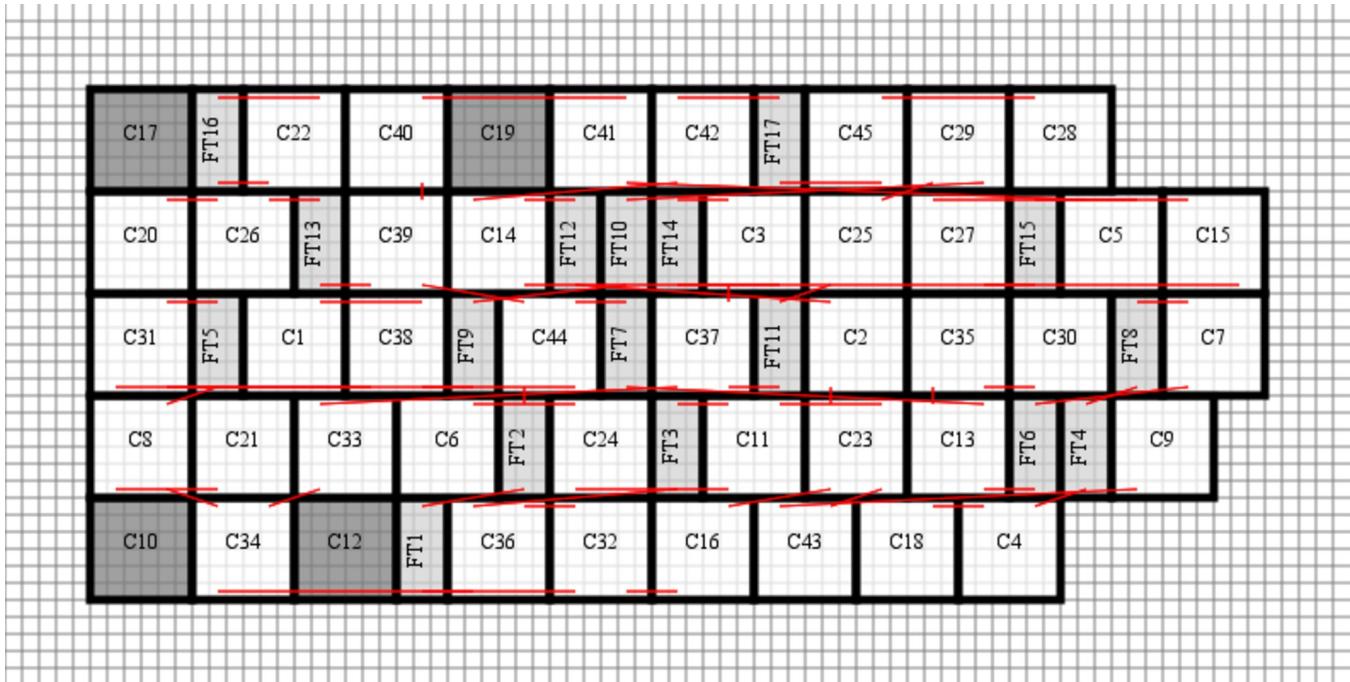


Figure 30: Benchmark 1 - Step 7/7 - Move feed-throughs to optimal location

4.2 Benchmark 1 - Final Routing

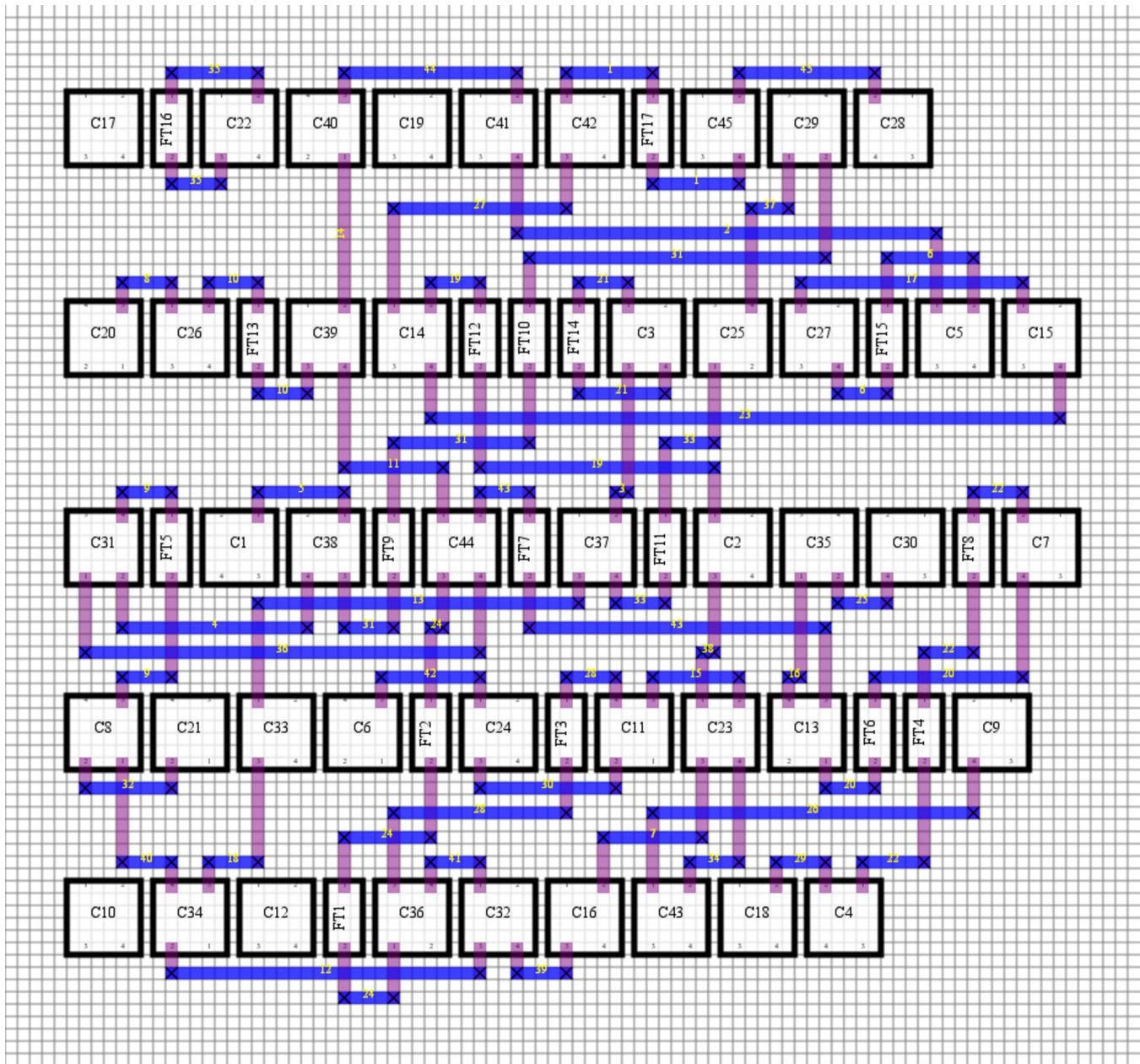


Figure 31: Benchmark 1 - Final routing

4.3 Benchmark 1 - Magic Screenshot

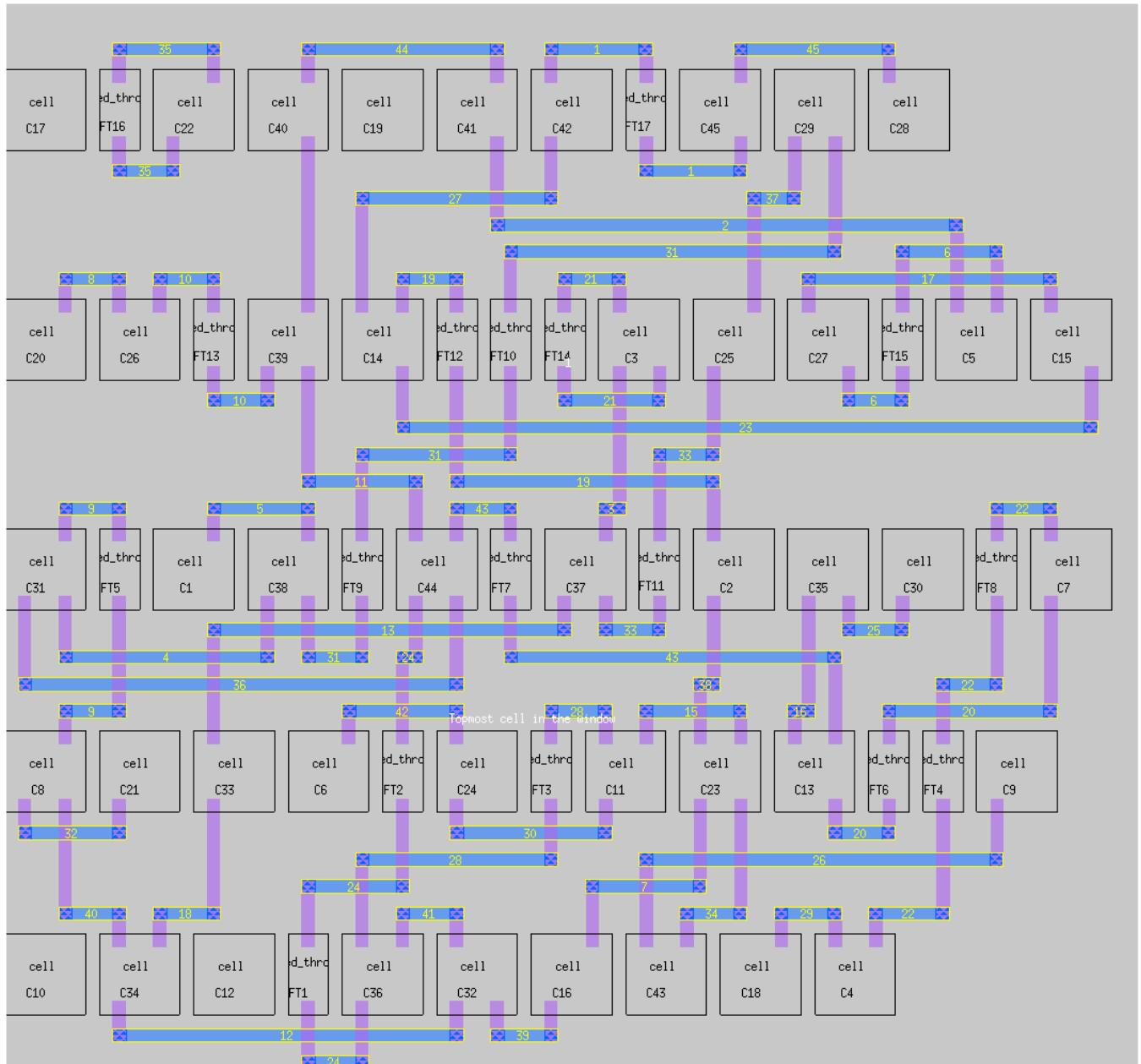


Figure 32: Benchmark 1 - Magic Screenshot

4.4 Benchmark 1 - Log file

```
1 Number of cell: 45
2 Number of nets: 45
3
4 Placement
5 _____
6 Initial grid size: 9 5
7 Number of feed through cells: 17
8
9 Routing
10 _____
11 Number of vias: 122
12 Total number of tracks: 21
13 Total wirelength: 1093
14
15 Size
16 _____
17 Total width: 82
18 Total height: 76
19 Total area: 6232
20 Squareness (width/height): 1.078947
21
22 Time
23 _____
24 Place time: 0.005293s
25 Route time: 0.000916s
26 Total time: 0.009209s
27
28 Memory used: 292.000000 kB
```

Listing 1: Benchmark 1 - Log

4.5 Benchmark 2 - Placement Steps

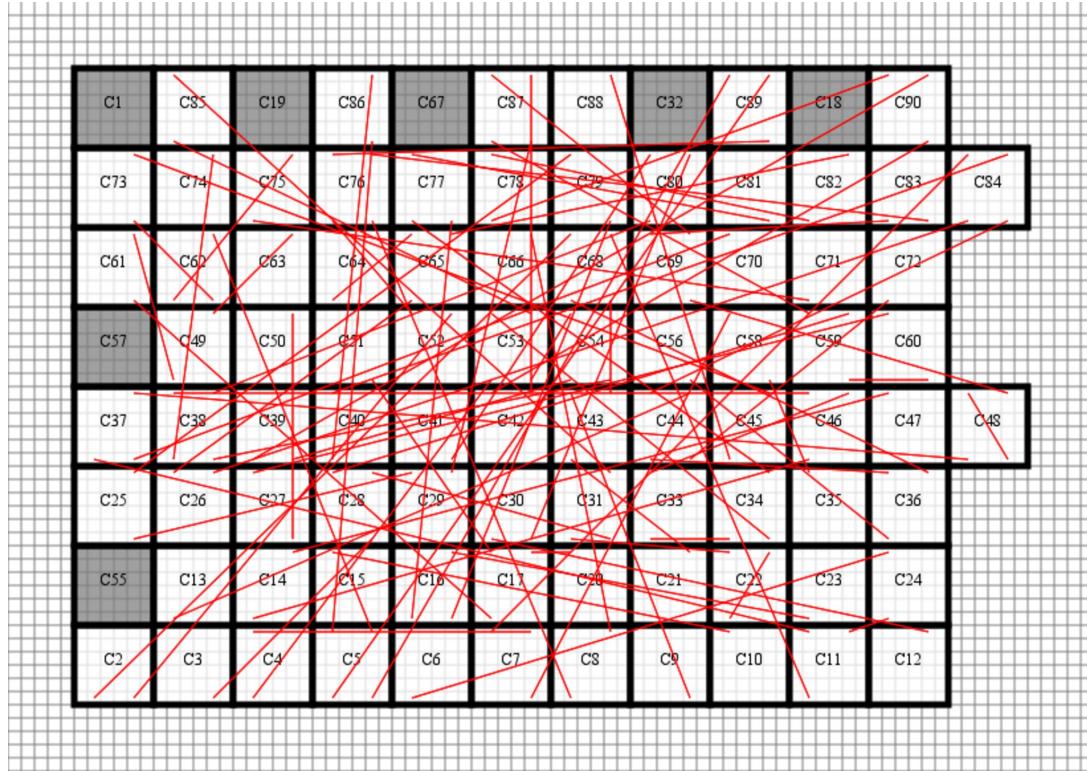


Figure 33: Benchmark 2 - Step 1/7 - Initial placement

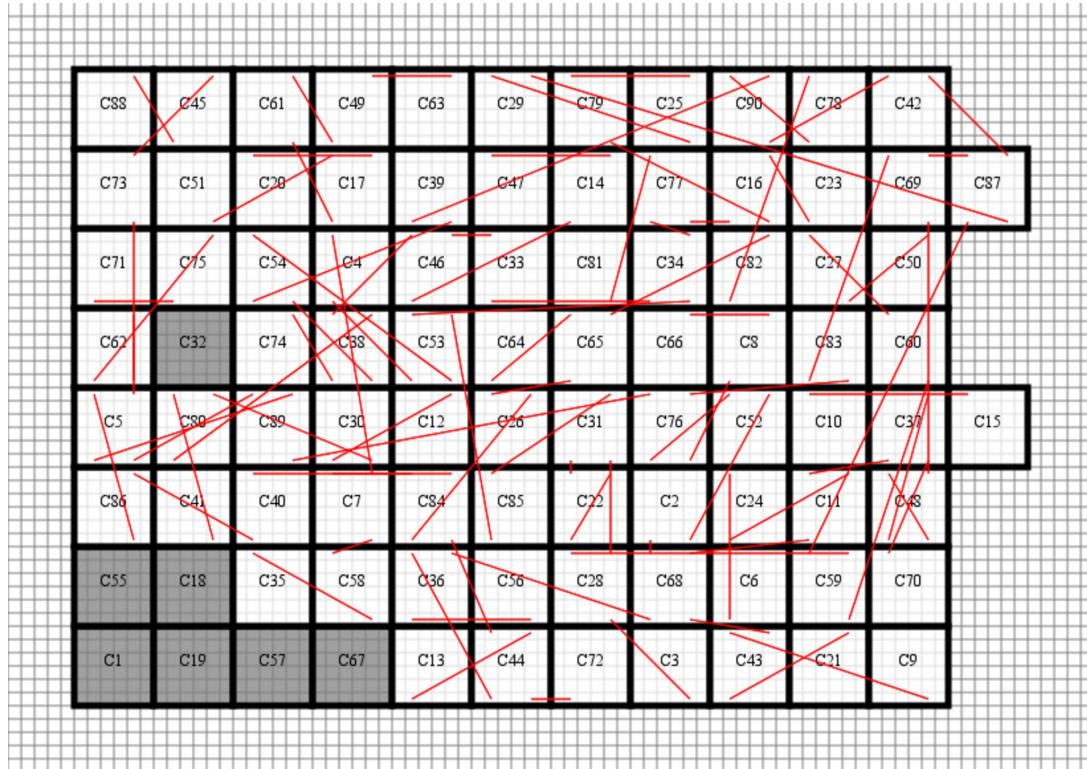


Figure 34: Benchmark 2 - Step 2/7 - Force directed placement

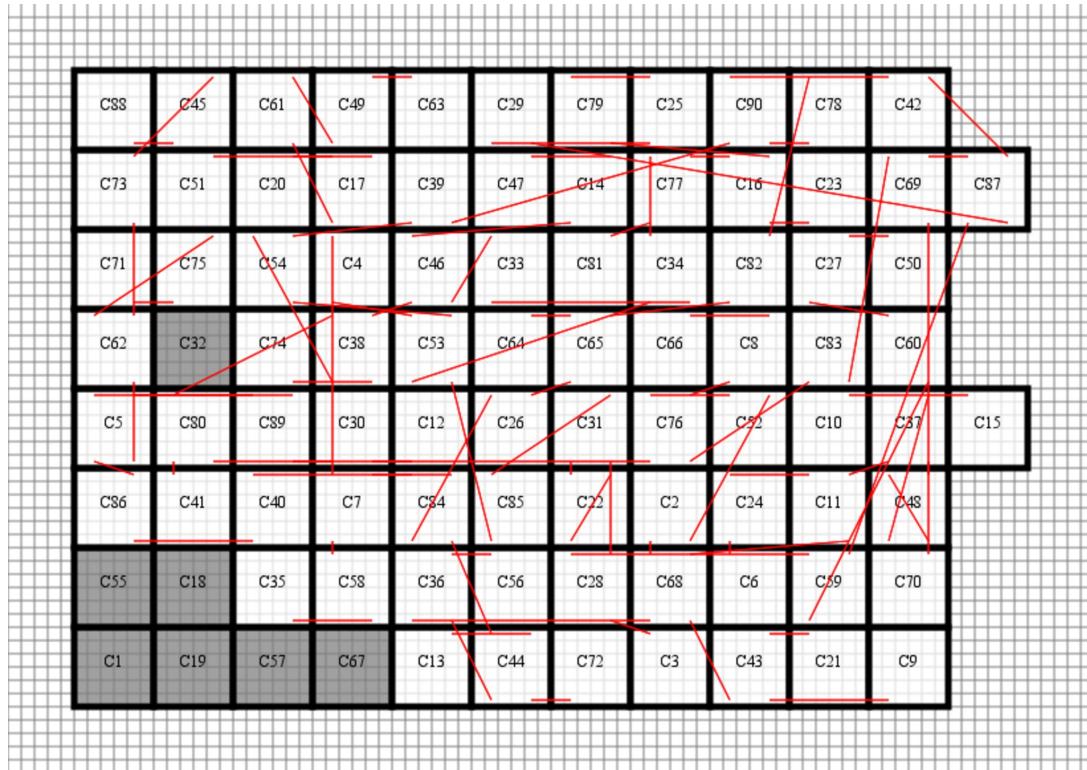


Figure 35: Benchmark 2 - Step 3/7 - Flip cells

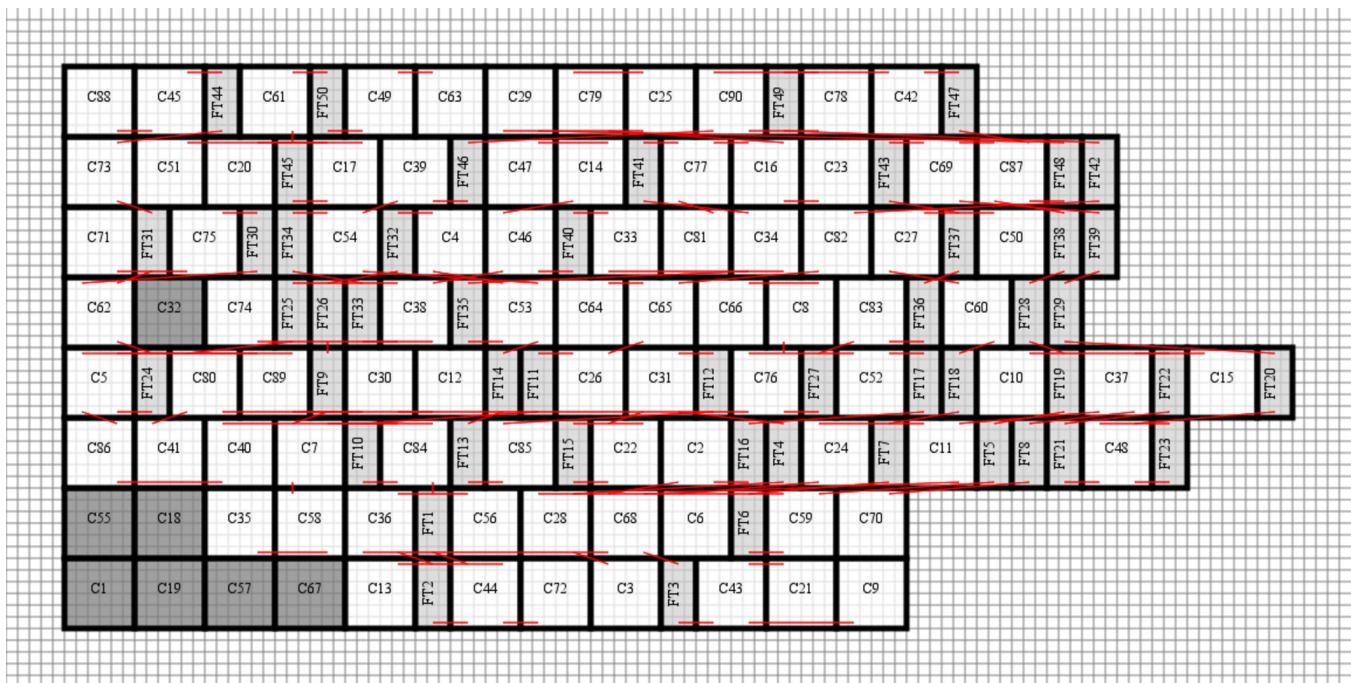


Figure 36: Benchmark 2 - Step 4/7 - Add feed-throughs

Figure 37: Benchmark 2 - Step 5/7 - Even out the row length

Figure 38: Benchmark 2 - Step 6/7 - Pull cells in the rows closer together

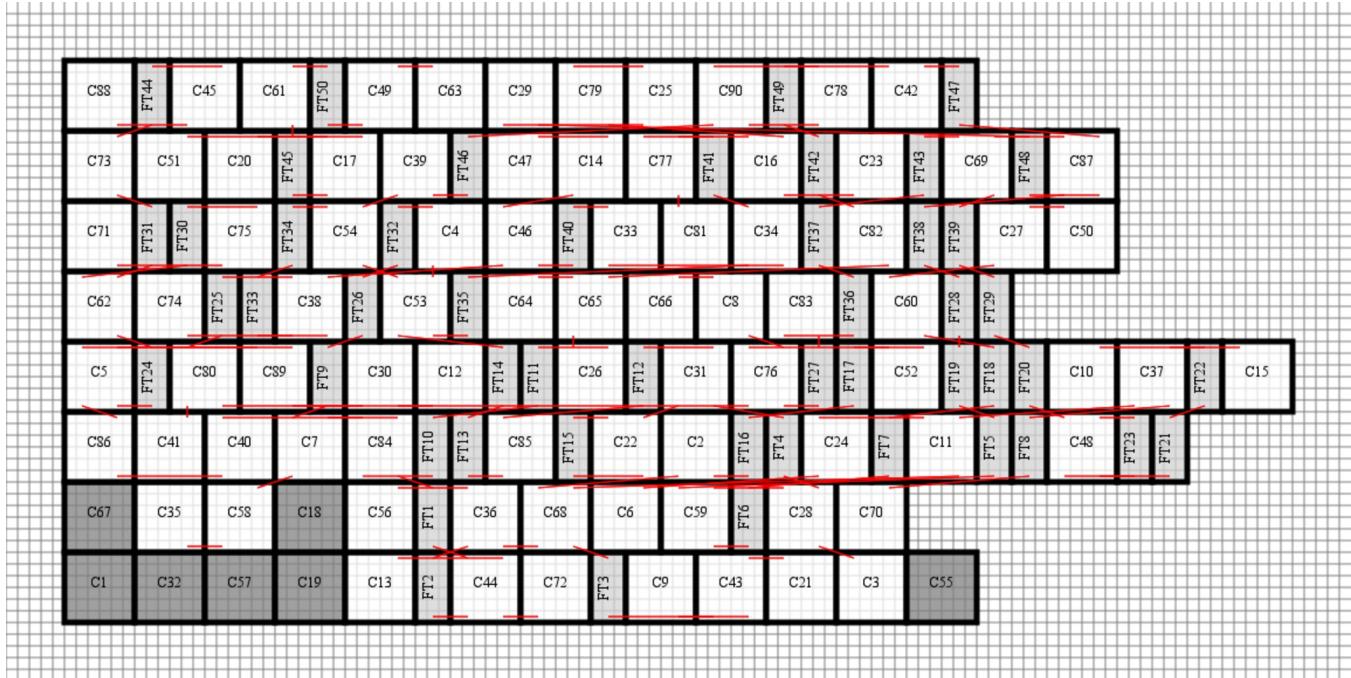


Figure 39: Benchmark 2 - Step 7/7 - Move feed-throughs to optimal location

4.6 Benchmark 2 - Final Routing

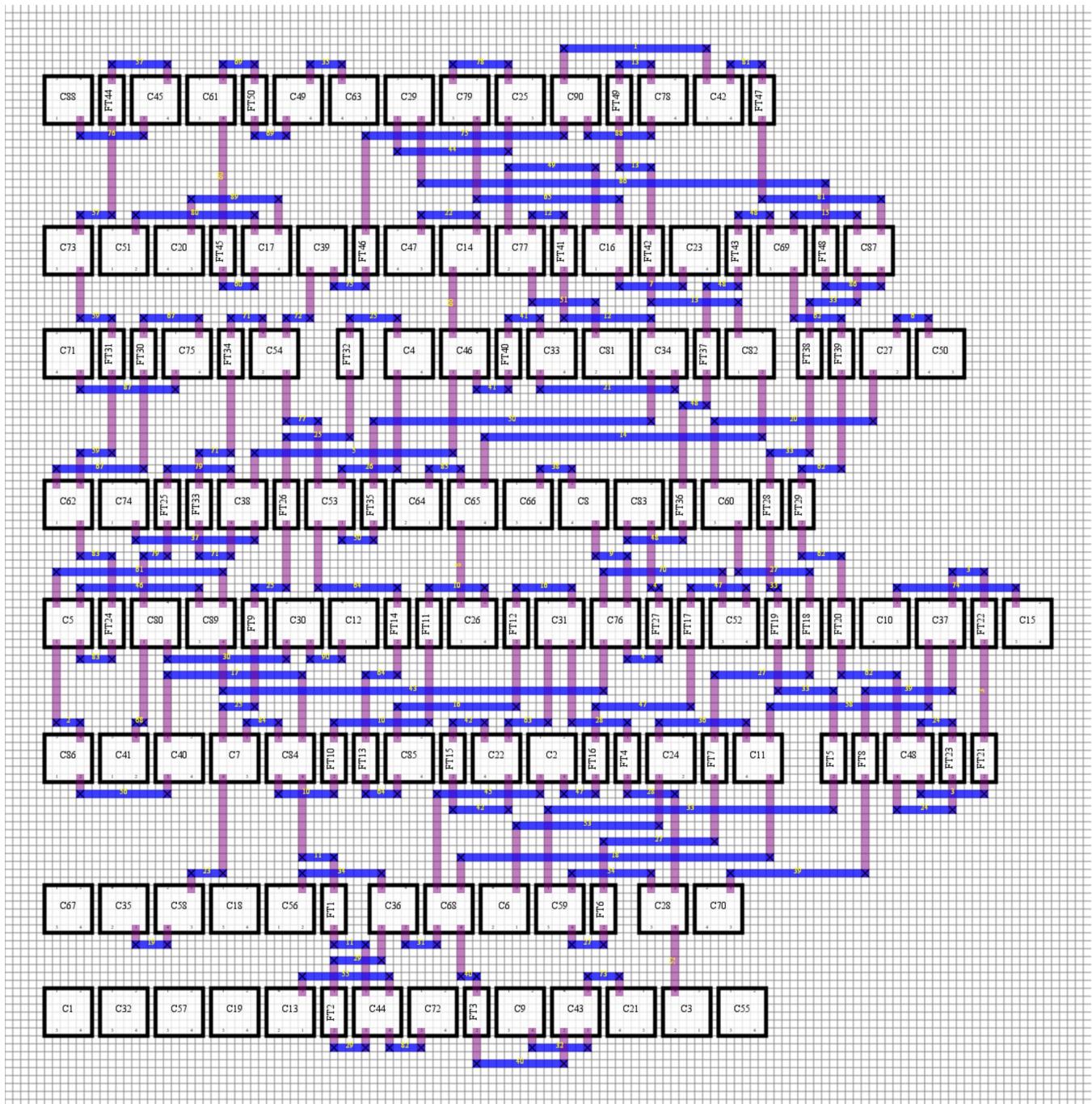


Figure 40: Benchmark 2 - Final routing

4.7 Benchmark 2 - Magic Screenshot

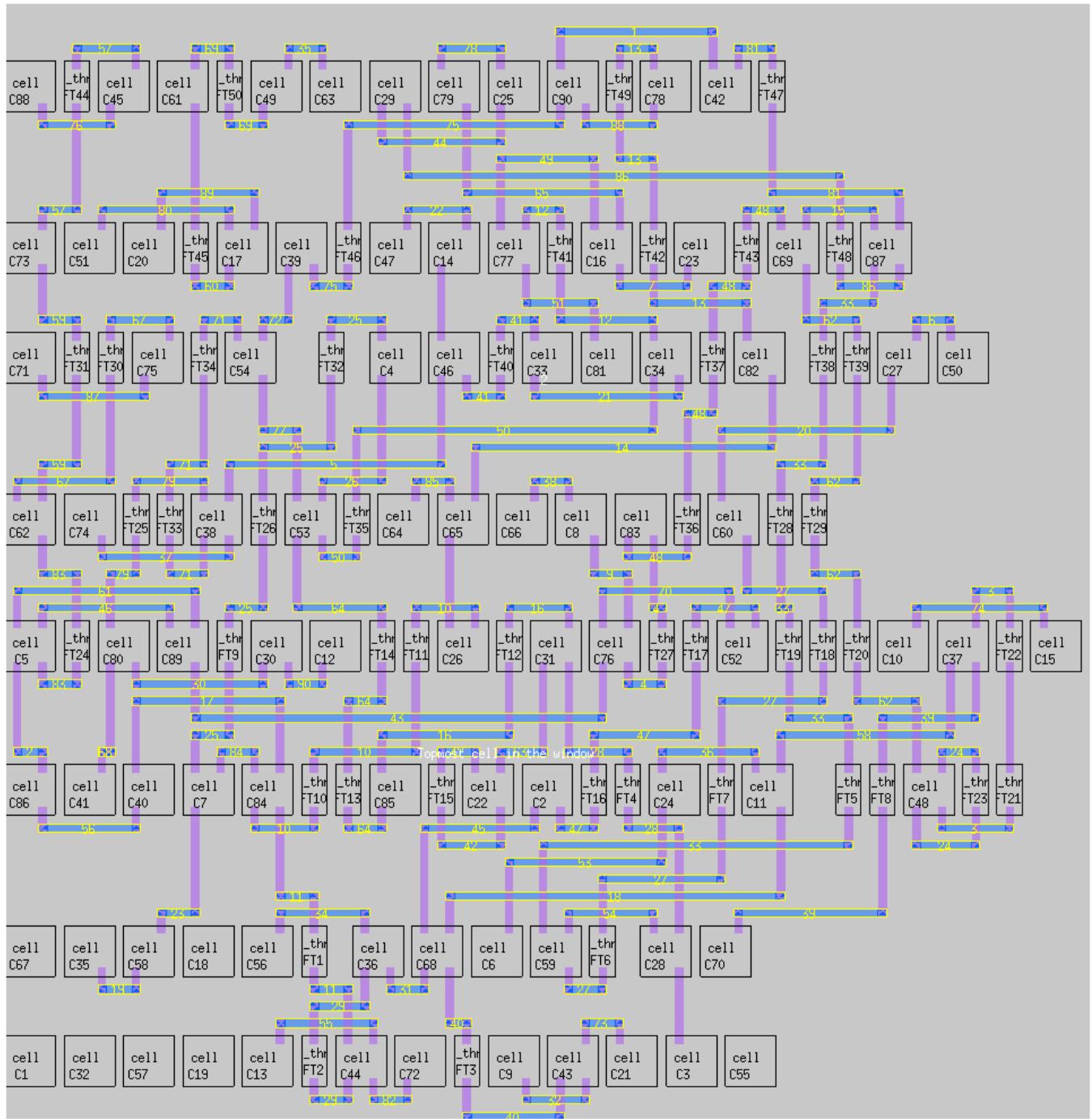


Figure 41: Benchmark 2 - Magic Screenshot

4.8 Benchmark 2 - Log file

```
1 Number of cell: 90
2 Number of nets: 90
3
4 Placement
5 _____
6 Initial grid size: 12 8
7 Number of feed through cells: 50
8
9 Routing
10 _____
11 Number of vias: 270
12 Total number of tracks: 37
13 Total wirelength: 2547
14
15 Size
16 _____
17 Total width: 127
18 Total height: 129
19 Total area: 16383
20 Squareness (width/height): 0.984496
21
22 Time
23 _____
24 Place time: 0.010814s
25 Route time: 0.002603s
26 Total time: 0.019800s
27
28 Memory used: 340.000000 kB
```

Listing 2: Benchmark 2 - Log

4.9 Benchmark 3 - Placement Steps

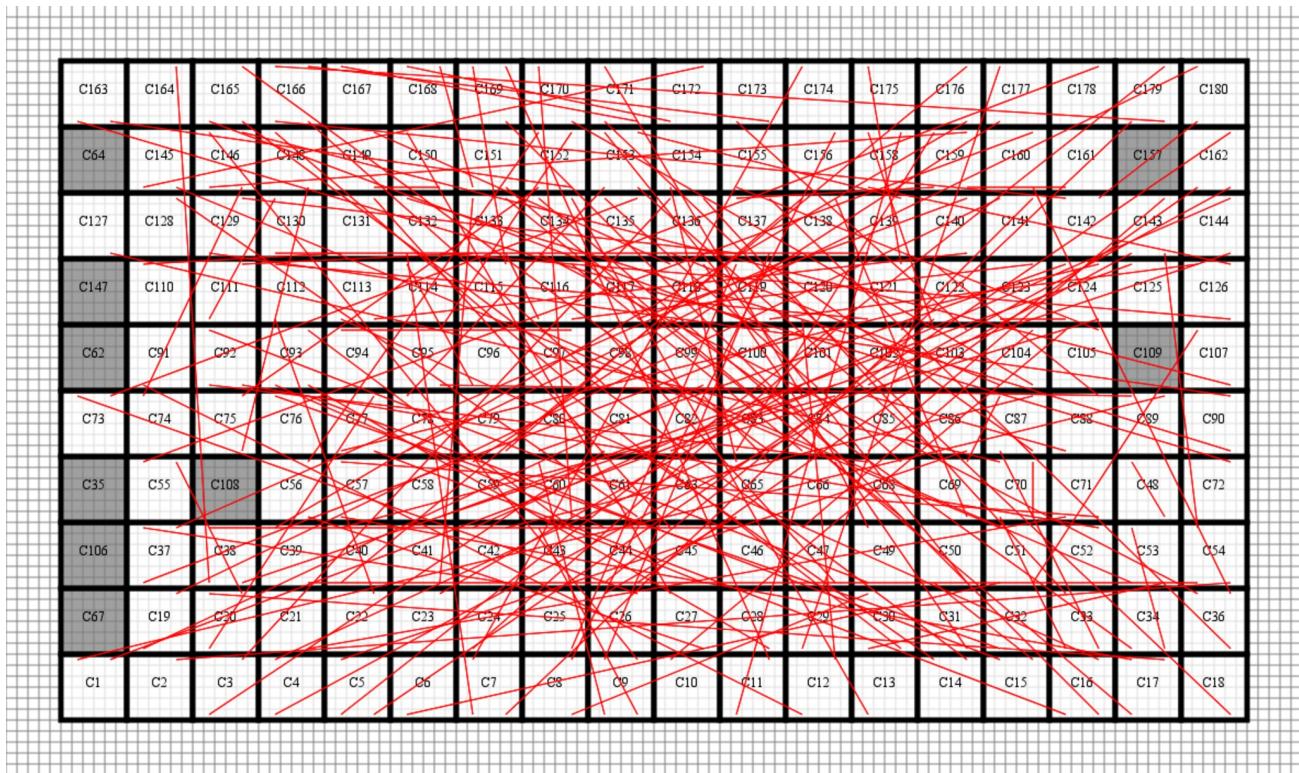


Figure 42: Benchmark 3 - Step 1/7 - Initial placement

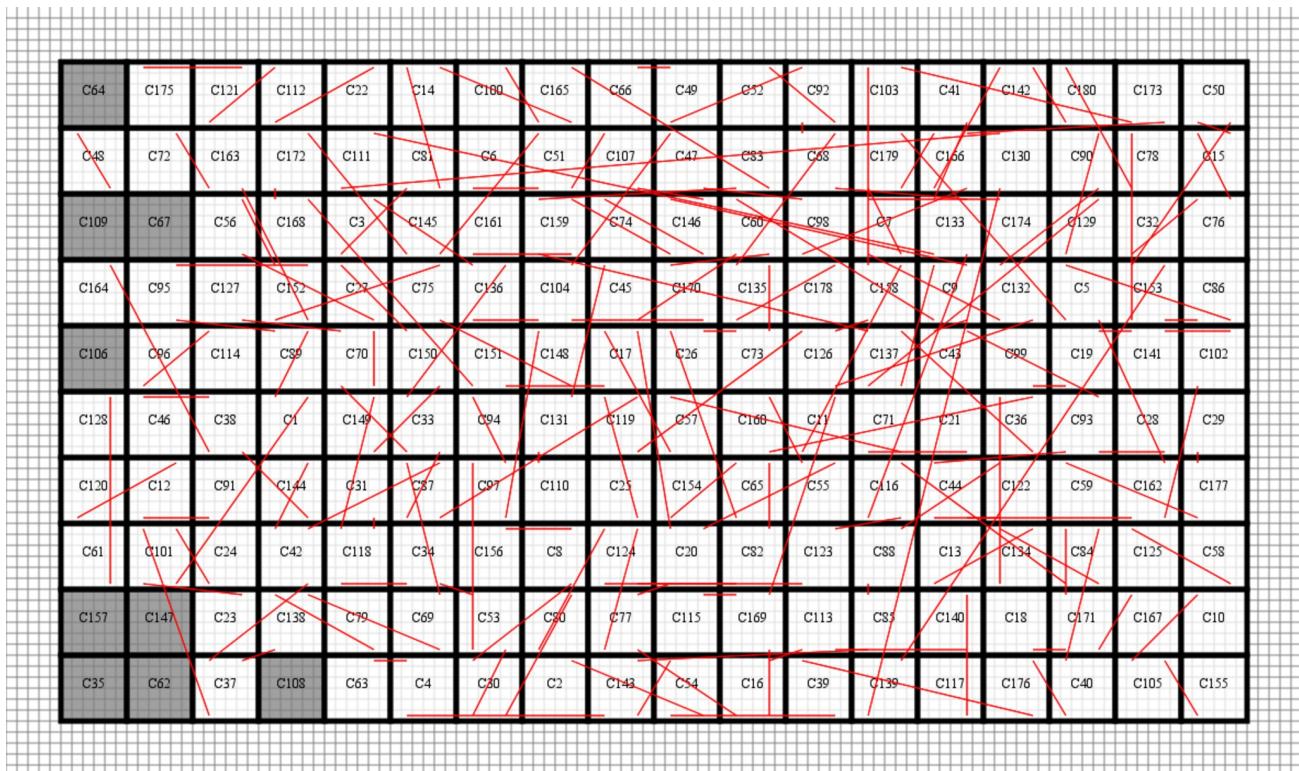


Figure 43: Benchmark 3 - Step 2/7 - Force directed placement

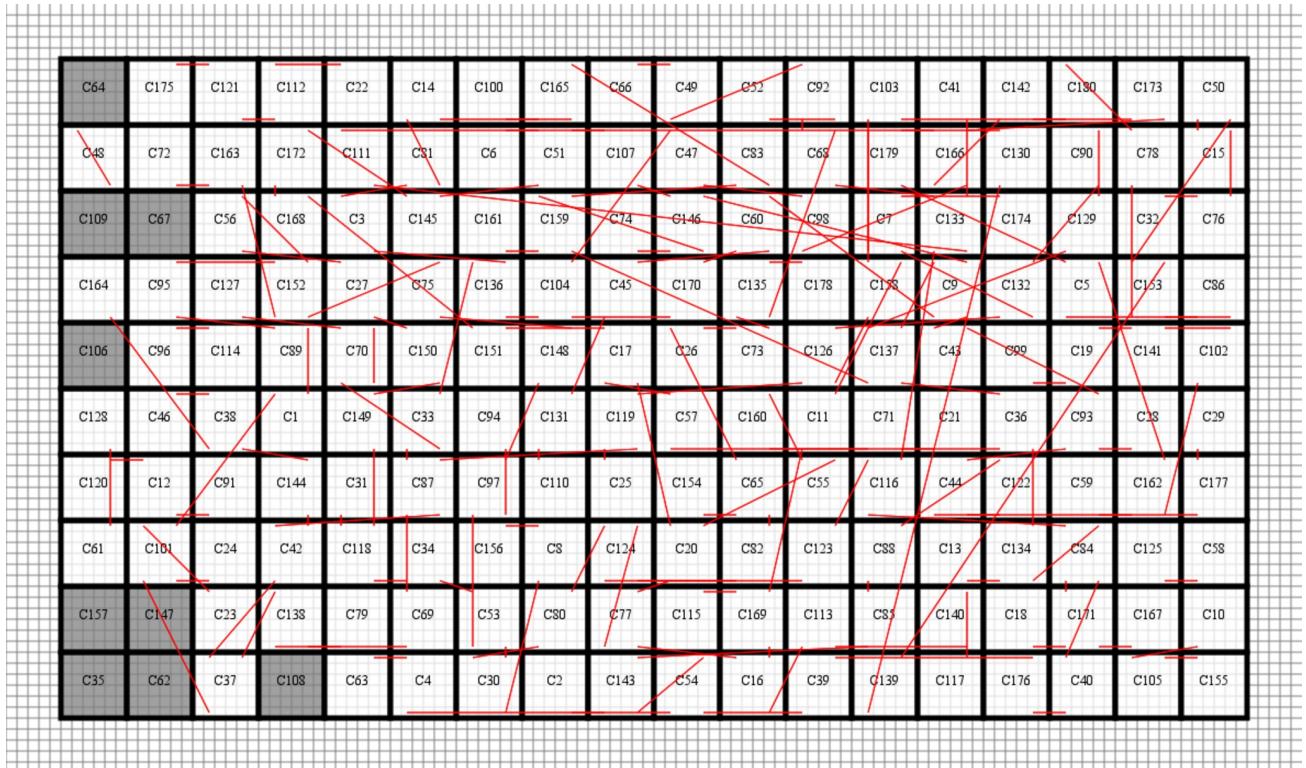


Figure 44: Benchmark 3- Step 3/7 - Flip cells

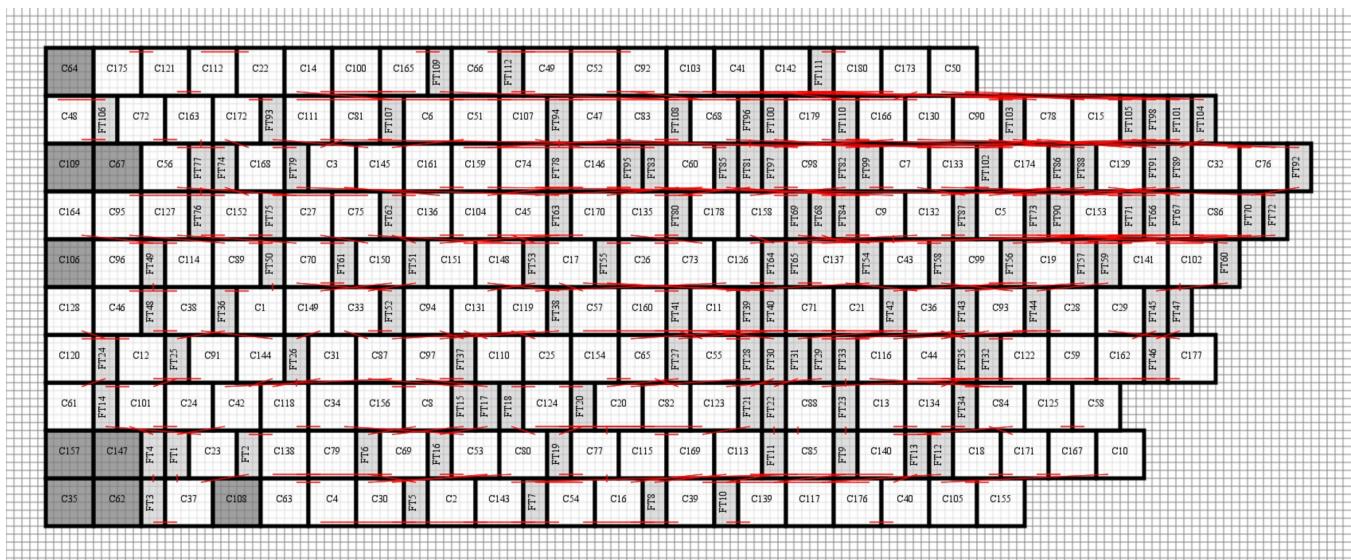


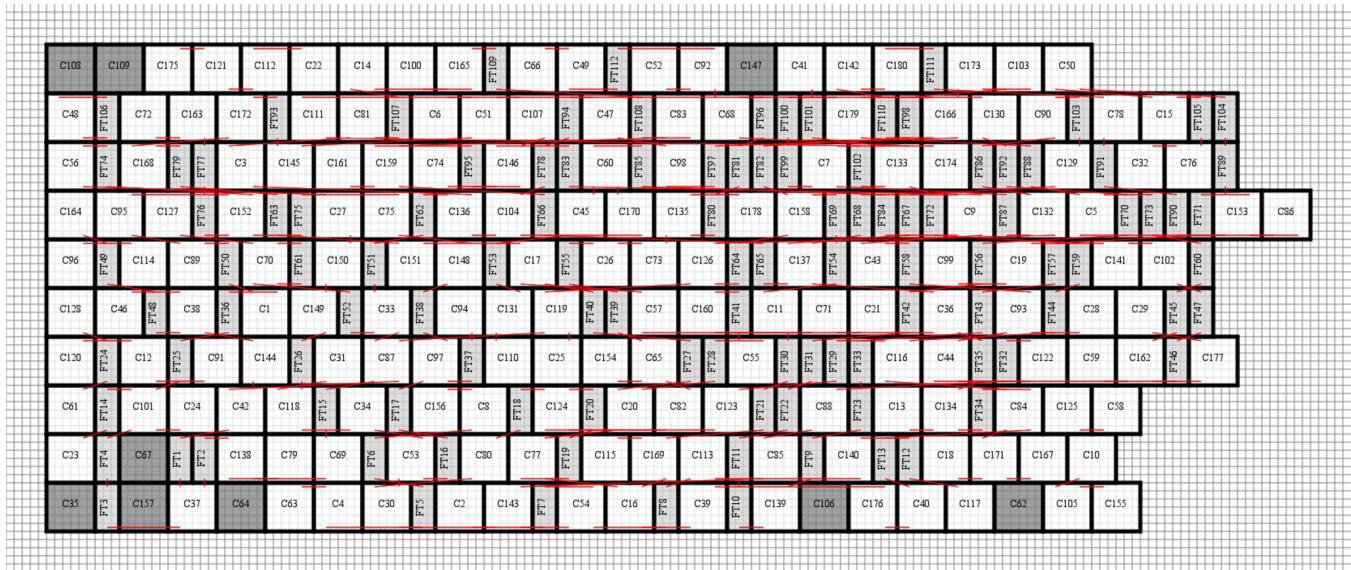
Figure 45: Benchmark 3- Step 4/7 - Add feed-throughs

C108	C175	C109	C121	C112	C22	C14	C100	C165	FT109	C66	C49	C52	C92	C103	C41	C142	FT111	C180	C173	C147	C50
C48	FT108	FT114	C72	C163	C172	FT93	C111	C81	FT107	C6	C51	C107	C74	FT94	C47	C83	FT108	C68	C130	C90	FT103
C56	FT117	FT114	C168	C169	C3	FT79	C145	C161	C159	C74	FT83	C146	C74	FT95	C60	FT85	FT81	C98	C133	C174	FT105
C164	C95	C127	C176	C152	C27	C75	FT75	C136	C104	C45	FT63	C104	C136	FT63	C170	C135	FT80	FT79	FT99	FT108	FT89
C96	FT149	C114	C89	FT70	C70	FT61	C150	FT31	C151	C148	FT33	C17	FT35	FT63	C173	C178	FT80	FT82	FT76	FT108	FT78
C128	C46	FT148	C38	FT36	C1	C149	C33	FT53	C94	C131	C119	C138	FT38	C57	C160	C166	FT110	C133	C174	C129	C153
C120	C46	FT14	C12	FT25	C91	C144	C31	C87	C97	C110	C25	C154	C65	FT41	C164	C166	FT102	C133	C174	C129	C153
C61	FT14	FT14	C101	C24	C42	C118	C34	C156	C8	FT15	FT37	C110	C20	C82	C124	C124	FT14	C137	C174	C132	FT105
C67	FT3	FT14	FT1	C23	FT2	C138	C79	FT16	C69	FT16	C53	C30	C143	FT19	C77	C115	C169	C113	C113	C166	FT105
C35	FT3	FT14	C157	C37	C64	C63	C4	C30	FT5	C2	FT7	C54	C16	FT8	C39	FT10	C139	C117	C176	C40	FT105

Figure 46: Benchmark 3- Step 5/7 - Even out the row length

C108	C109	C175	C121	C112	C22	C14	C100	C165	FT109	C66	C49	C52	C92	C147	C41	C142	FT111	C180	C173	C103	C50
C48	FT108	FT114	C72	C163	C172	FT93	C111	C81	FT107	C6	C51	C107	C74	FT94	C47	C83	FT108	C68	C130	C90	FT103
C56	FT117	FT114	C168	C169	C3	FT79	C145	C161	C159	C74	FT83	C146	C74	FT95	C60	FT85	FT81	C98	C133	C174	FT105
C164	C95	C127	C176	C152	C27	C75	FT75	C136	C104	C45	FT63	C104	C136	FT63	C170	C135	FT80	FT79	FT99	FT108	FT78
C96	FT149	C114	C89	FT70	C70	FT61	C150	FT31	C151	C148	FT33	C17	FT35	FT63	C173	C178	FT80	FT82	FT76	FT108	FT78
C128	C46	FT148	C38	FT36	C1	C149	C33	FT53	C94	C131	C119	C138	FT38	C57	C160	C164	FT41	C137	C174	C132	FT105
C120	C46	FT14	C12	FT25	C91	C144	FT26	C31	C87	C97	FT37	C110	C25	C154	C65	FT41	C164	C166	C133	C174	C132
C61	FT14	FT14	C101	C24	C42	C118	C34	C156	C8	FT15	FT37	C110	C20	C82	C124	C124	FT14	C137	C174	C132	FT105
C33	FT3	FT14	C67	FT1	FT2	C138	C79	FT16	C69	FT16	C53	C30	C143	FT19	C77	C115	C169	C113	C113	C166	FT105
C35	FT3	FT14	C157	C37	C64	C63	C4	C30	FT5	C2	FT7	C54	C16	FT8	C39	FT10	C139	C117	C176	C40	FT105

Figure 47: Benchmark 3- Step 6/7 - Pull cells in the rows closer together

**Figure 48:** Benchmark 3- Step 7/7 - Move feed-throughs to optimal location

4.10 Benchmark 3 - Final Routing

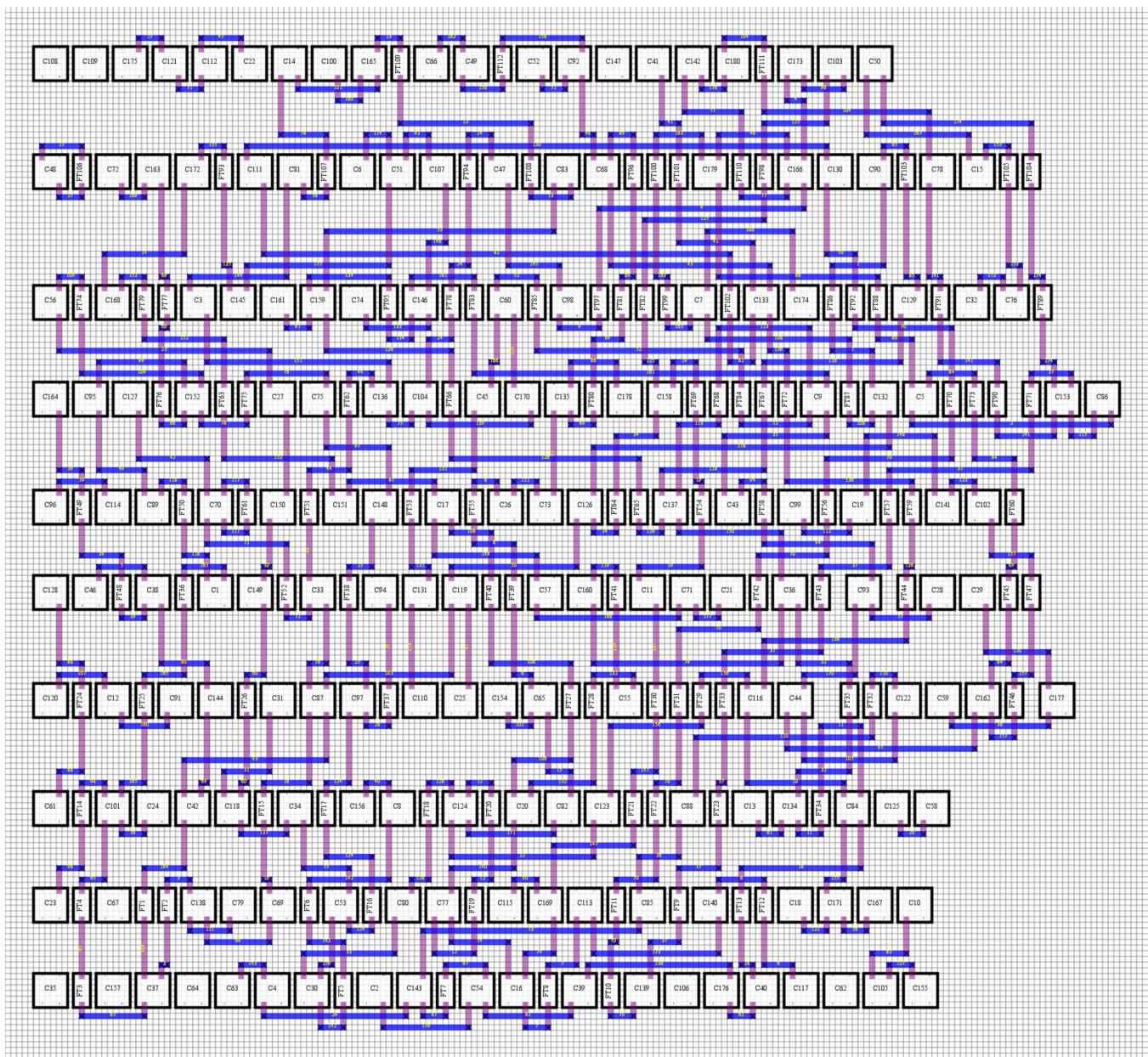


Figure 49: Benchmark 3 - Final routing

4.11 Benchmark 3 - Magic Screenshot



Figure 50: Benchmark 3- Magic Screenshot

4.12 Benchmark 3 - Log file

```
1 Number of cell: 180
2 Number of nets: 180
3
4 Placement
5 _____
6 Initial grid size: 18 10
7 Number of feed through cells: 112
8
9 Routing
10 _____
11 Number of vias: 562
12 Total number of tracks: 53
13 Total wirelength: 6114
14
15 Size
16 _____
17 Total width: 191
18 Total height: 175
19 Total area: 33425
20 Squareness (width/height): 1.091429
21
22 Time
23 _____
24 Place time: 0.023764s
25 Route time: 0.007911s
26 Total time: 0.044868s
27
28 Memory used: 432.000000 kB
```

Listing 3: Benchmark 3 - Log

4.13 Benchmark 4 - Placement Steps

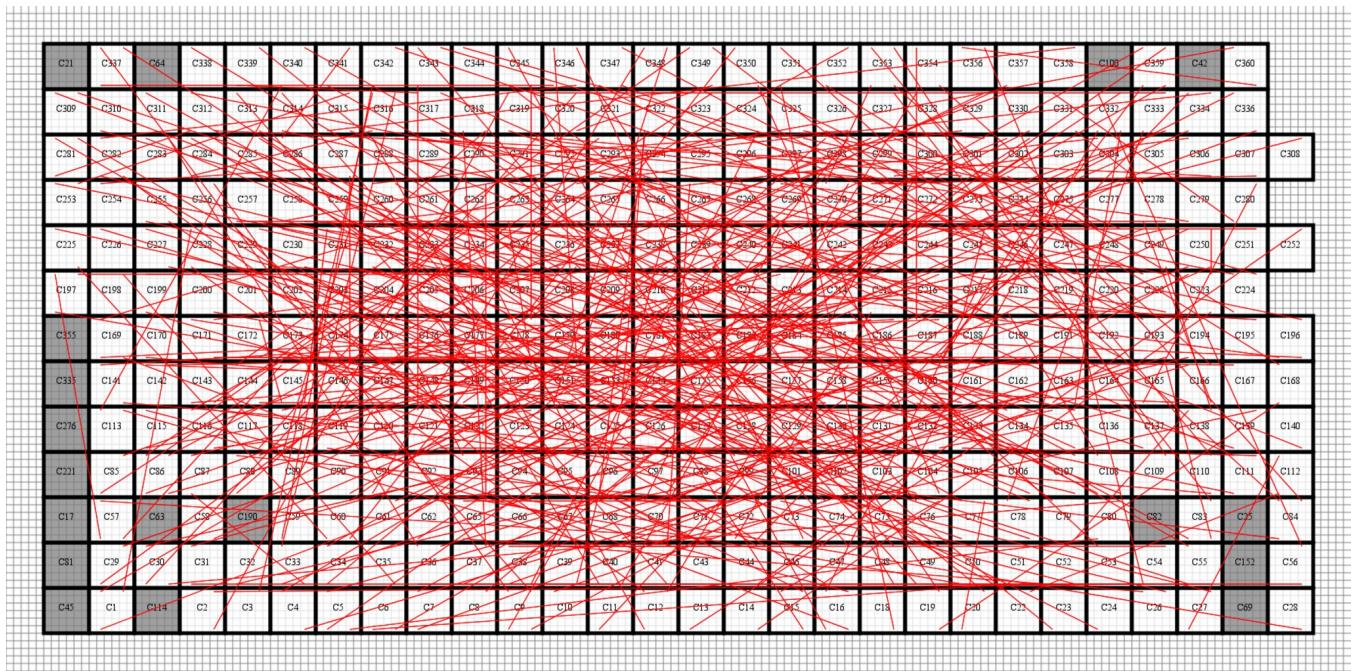


Figure 51: Benchmark 4 - Step 1/7 - Initial placement

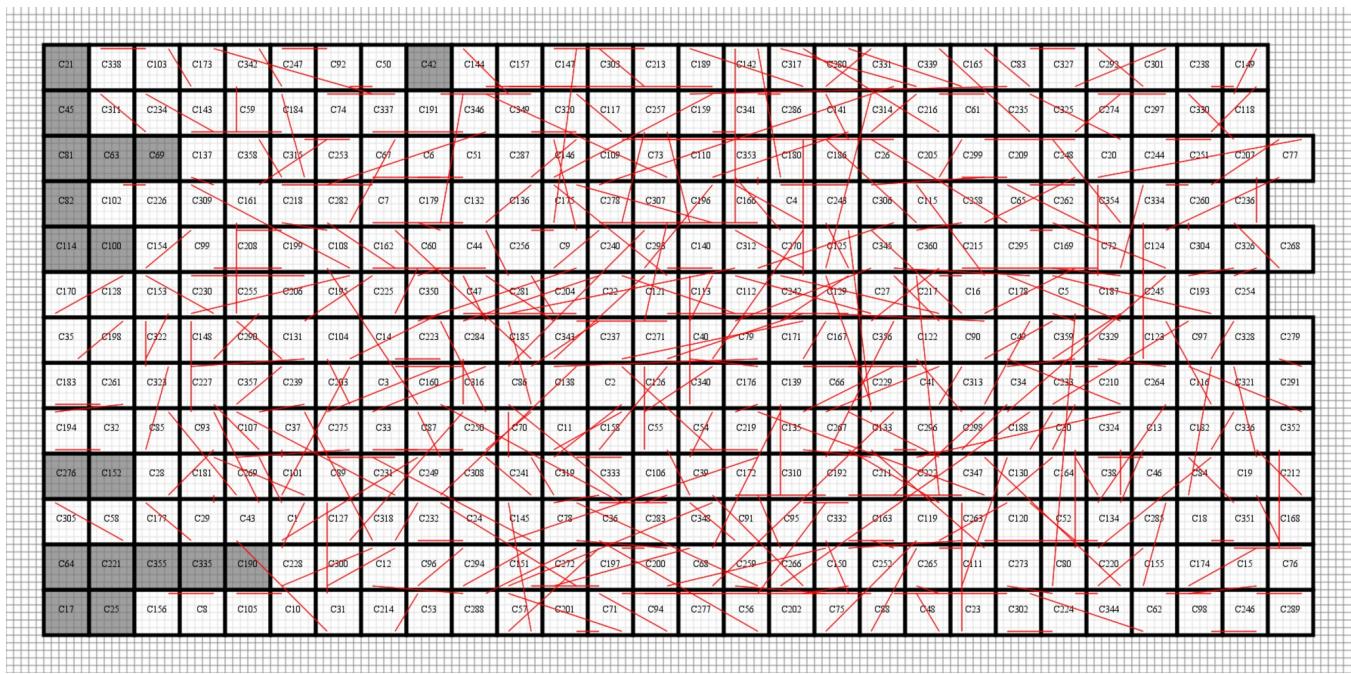


Figure 52: Benchmark 4 - Step 2/7 - Force directed placement

C21	C338	C103	C173	C342	C247	C92	C50	C42	C144	C157	C147	C303	C213	C189	C142	C317	C280	C331	C339	C145	C33	C327	C329	C301	C238	C149	
C45	C311	C134	C143	C59	C184	C74	C337	C191	C346	C549	C330	C117	C257	C159	C341	C286	C141	C314	C216	C61	C235	C325	C274	C397	C330	C118	
C81	C63	C69	C137	C318	C119	C253	C67	C6	C51	C287	C146	C109	C73	C116	C353	C180	C186	C206	C35	C399	C309	C248	C20	C244	C251	C301	C177
C82	C102	C126	C309	C161	C218	C162	C7	C179	C132	C136	C175	C78	C307	C196	C166	C4	C243	C306	C115	C258	C65	C262	C354	C334	C360	C236	
C114	C100	C154	C99	C208	C199	C108	C162	C60	C44	C256	C9	C340	C293	C140	C312	C270	C125	C345	C280	C215	C295	C169	C72	C124	C304	C326	C368
C170	C128	C153	C230	C255	C306	C195	C225	C358	C47	C31	C204	C22	C121	C113	C112	C242	C129	C27	C217	C16	C178	C5	C187	C245	C193	C254	
C35	C198	C322	C148	C290	C131	C104	C14	C23	C284	C185	C141	C237	C271	C30	C79	C171	C167	C356	C122	C90	C49	C359	C329	C123	C97	C328	C279
C183	C361	C323	C227	C357	C239	C303	C3	C160	C316	C36	C138	C3	C126	C340	C176	C139	C56	C229	C41	C313	C34	C233	C10	C264	C116	C321	C391
C194	C32	C35	C93	C107	C37	C275	C33	C37	C380	C70	C11	C138	C55	C54	C219	C135	C67	C133	C396	C48	C188	C50	C324	C13	C32	C336	C352
C276	C152	C28	C181	C269	C101	C39	C23	C249	C308	C241	C319	C333	C106	C39	C172	C310	C192	C214	C222	C347	C130	C164	C38	C46	C34	C19	C212
C305	C58	C177	C29	C43	C1	C127	C318	C232	C24	C145	C78	G36	C283	C348	G89	C95	C332	C163	C119	C263	C120	C52	C134	C387	C18	C351	C168
C64	C221	C355	C335	C190	C228	C300	C12	C96	C294	C161	C272	C197	C300	C68	C159	C36	C150	C252	C265	C111	C273	C80	C220	C155	C174	C15	C76
C17	C25	C156	C8	C105	C10	C31	C14	C53	C388	C57	C301	C7	C94	C277	C56	C302	C75	C58	C23	C302	C224	C344	C62	C98	C246	C389	

Figure 53: Benchmark 4 - Step 3/7 - Flip cells

C1	C38	C93	C13	C340	C251	C47	C92	C50	C42	C144	C131	C210	C147	C303	C213	C189	C142	C317	C280	C331	C339	C145	C33	C327	C329	C301	C238	C149
C45	C311	C134	C143	C59	C184	C74	C337	C191	C346	C549	C330	C117	C257	C159	C341	C286	C141	C314	C216	C61	C235	C325	C274	C397	C330	C118		
C81	C63	C69	C137	C318	C318	C119	C253	C67	C6	C51	C287	C146	C109	C73	C116	C353	C180	C186	C206	C35	C399	C309	C248	C20	C244	C251	C301	C177
C82	C102	C128	C309	C161	C134	C7	C218	C162	C60	C44	C256	C9	C340	C293	C140	C312	C270	C125	C345	C280	C215	C295	C169	C72	C124	C304	C326	C368
C114	C100	C154	C99	C208	C199	C108	C162	C60	C44	C256	C9	C340	C293	C140	C312	C270	C125	C345	C280	C215	C295	C169	C72	C124	C304	C326	C368	
C170	C128	C153	C230	C255	C306	C195	C225	C358	C47	C31	C204	C22	C121	C113	C112	C242	C129	C27	C217	C16	C178	C5	C187	C245	C193	C254		
C35	C198	C322	C148	C290	C131	C104	C14	C23	C284	C185	C141	C237	C271	C30	C79	C171	C167	C356	C122	C90	C49	C359	C329	C123	C97	C328	C279	
C183	C361	C323	C227	C357	C239	C303	C3	C160	C316	C36	C138	C3	C126	C340	C176	C139	C56	C229	C41	C313	C34	C233	C10	C264	C116	C321	C391	
C194	C32	C35	C93	C107	C37	C275	C33	C37	C380	C70	C11	C138	C55	C54	C219	C135	C67	C133	C396	C48	C188	C50	C324	C13	C32	C336	C352	
C276	C152	C28	C181	C269	C101	C39	C23	C249	C308	C241	C319	C333	C106	C39	C172	C310	C192	C214	C222	C347	C130	C164	C38	C46	C34	C19	C212	
C305	C58	C177	C29	C43	C1	C127	C318	C232	C24	C145	C78	G36	C283	C348	G89	C95	C332	C163	C119	C263	C120	C52	C134	C387	C18	C351	C168	
C64	C221	C355	C335	C190	C228	C300	C12	C96	C294	C161	C272	C197	C300	C68	C159	C36	C150	C252	C265	C111	C273	C80	C220	C155	C174	C15	C76	
C17	C25	C156	C8	C105	C10	C31	C14	C53	C388	C57	C301	C7	C94	C277	C56	C302	C75	C58	C23	C302	C224	C344	C62	C98	C246	C389		

Figure 54: Benchmark 4 - Step 4/7 - Add feed-throughs

C1	C38	C93	C13	C340	C251	C47	C92	C50	C42	C144	C131	C210	C147	C303	C213	C189	C142	C317	C280	C331	C339	C145	C33	C327	C329	C301	C238	C149
C45	C311	C134	C143	C59	C184	C74	C337	C191	C346	C549	C330	C117	C257	C159	C341	C286	C141	C314	C216	C61	C235	C325	C274	C397	C330	C118		
C81	C63	C69	C137	C318	C318	C119	C253	C67	C6	C51	C287	C146	C109	C73	C116	C353	C180	C186	C206	C35	C399	C309	C248	C20	C244	C251	C301	C177
C82	C102	C128	C309	C161	C134	C7	C218	C162	C60	C44	C256	C9	C340	C293	C140	C312	C270	C125	C345	C280	C215	C295	C169	C72	C124	C304	C326	C368
C114	C100	C154	C99	C208	C199	C108	C162	C60	C44	C256	C9	C340	C293	C140	C312	C270	C125	C345	C280	C215	C295	C169	C72	C124	C304	C326	C368	
C170	C128	C153	C230	C255	C306	C195	C225	C358	C47	C31	C204	C22	C121	C113	C112	C242	C129	C27	C217	C16	C178	C5	C187	C245	C193	C254		
C35	C198	C322	C148	C290	C131	C104	C14	C23	C284	C185	C141	C237	C271	C30	C79	C171	C167	C356	C122	C90	C49	C359	C329	C123	C97	C328	C279	
C183	C361	C323	C227	C357	C239	C303	C3	C160	C316	C36	C138	C3	C126	C340	C176	C139	C56	C229	C41	C313	C34	C233	C10	C264	C116	C321	C391	
C194	C32	C35	C93	C107	C37	C275	C33	C37	C380	C70	C11	C138	C55	C54	C219	C135	C67	C133	C396	C48	C188	C50	C324	C13	C32	C336	C352	
C276	C152	C28	C181	C269	C101	C39	C23	C249	C308	C241	C319	C333	C106	C39	C172	C310	C192	C214	C222	C347	C130	C164	C38	C46	C34	C19	C212	
C305	C58	C177	C29	C43	C1	C127	C318	C232	C24	C145	C78	G36	C283	C348	G89	C95	C332	C163	C119	C263	C120	C52	C134	C387	C18	C351	C168	
C64	C221	C355	C335	C190	C228	C300	C12	C96	C294	C161	C272	C197	C300	C68	C159	C36	C150	C252	C265	C111	C273	C80	C220	C155	C174	C15	C76	
C17	C25	C156	C8	C105	C10	C31	C14	C53	C388	C57	C301	C7	C94	C277	C56	C302	C75	C58	C23	C302	C224	C344	C62	C98	C246	C389		

Figure 55: Benchmark 4 - Step 5/7 - Even out the row length

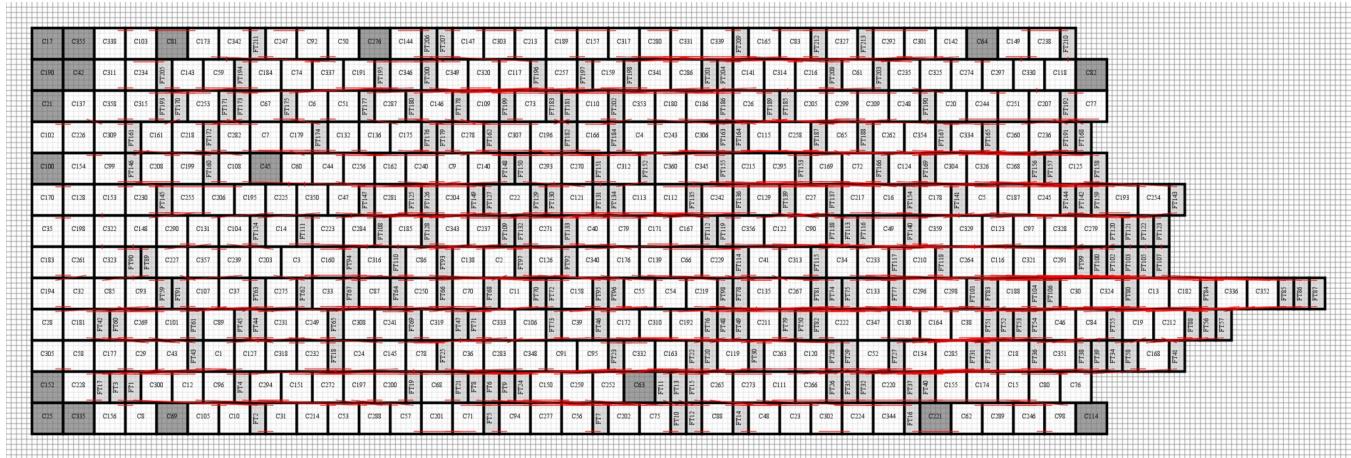


Figure 56: Benchmark 4 - Step 6/7 - Pull cells in the rows closer together

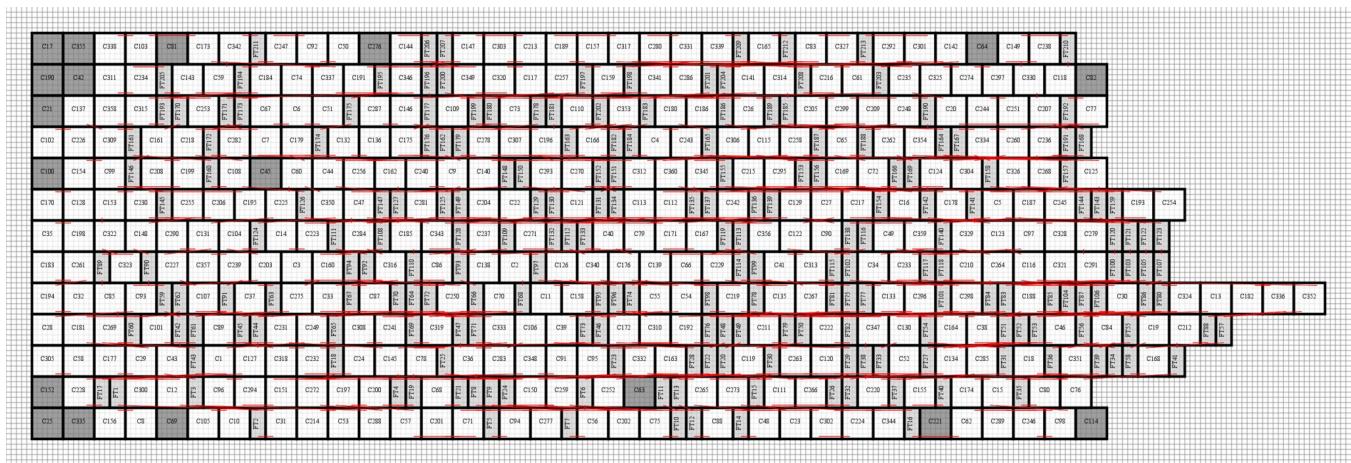


Figure 57: Benchmark 4 - Step 7/7 - Move feed-throughs to optimal location

4.14 Benchmark 4 - Final Routing

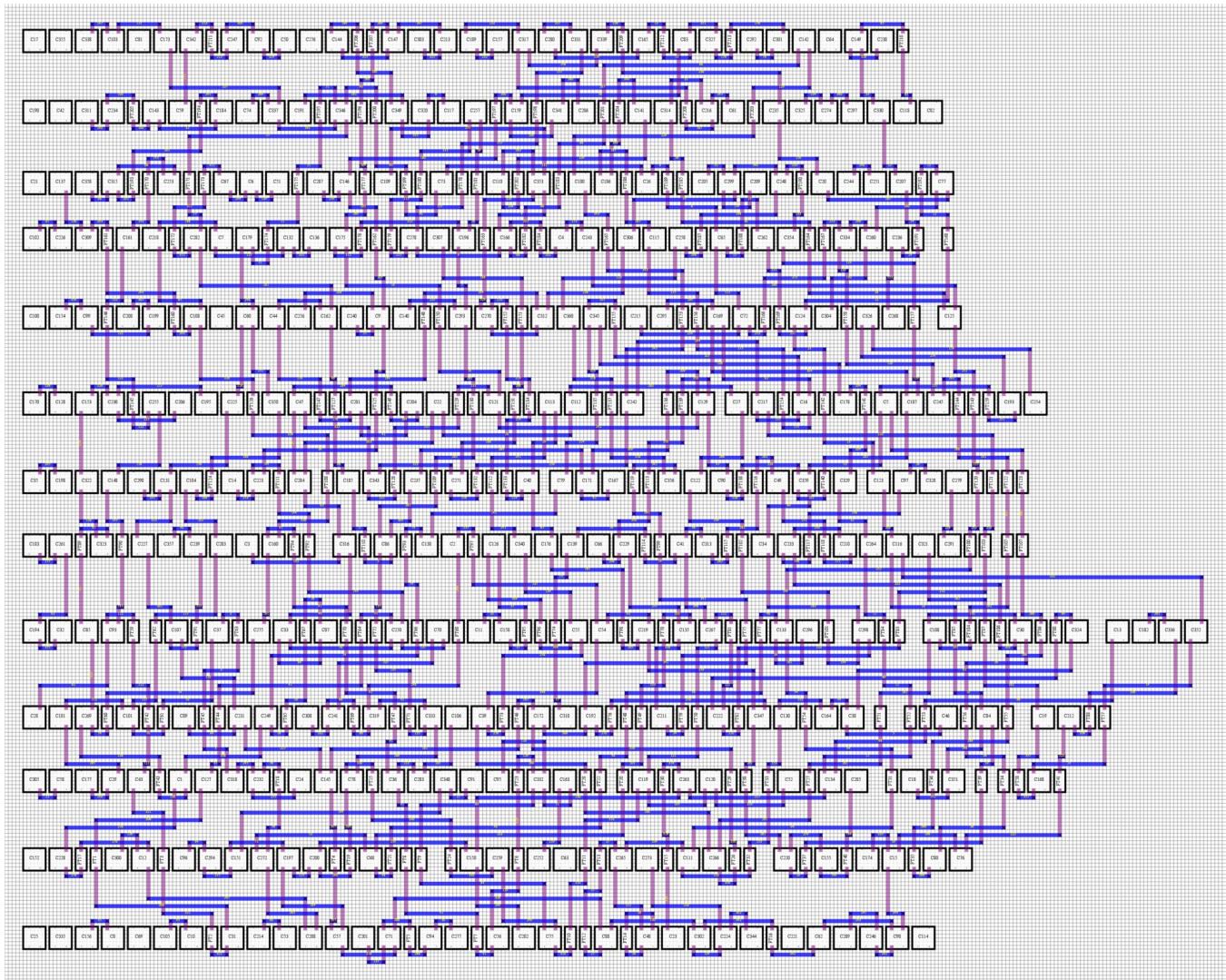


Figure 58: Benchmark 4 - Final routing

4.15 Benchmark 4 - Magic Screenshot

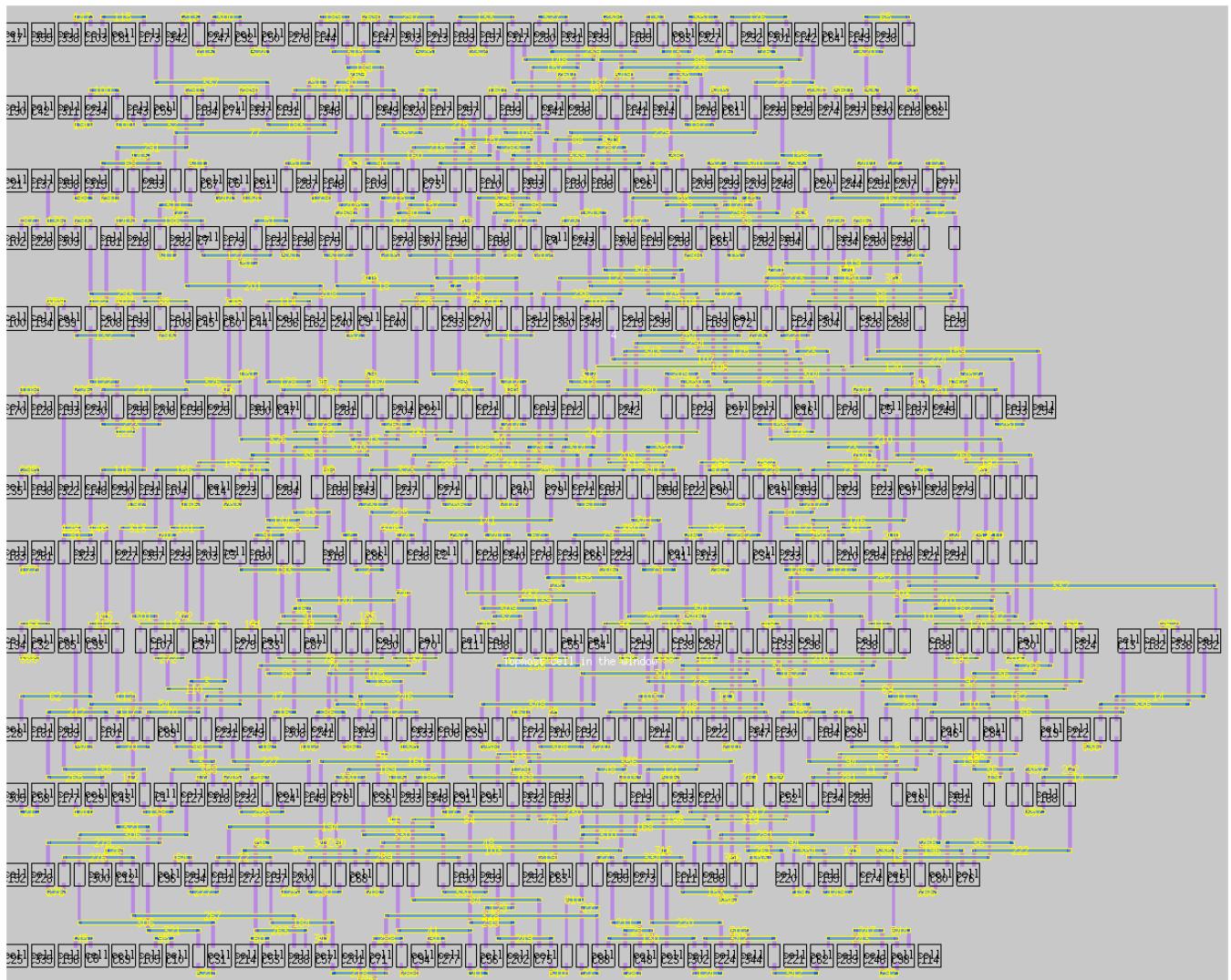


Figure 59: Benchmark 4 - Magic Screenshot

4.16 Benchmark 4 - Log file

```
1 Number of cell: 360
2 Number of nets: 360
3
4 Placement
5
6 Initial grid size: 28 13
7 Number of feed through cells: 213
8
9 Routing
10
11 Number of vias: 1126
12 Total number of tracks: 81
13 Total wirelength: 14108
14
15 Size
16
17 Total width: 317
18 Total height: 252
19 Total area: 79884
20 Squareness (width/height): 1.257937
21
22 Time
23
24 Place time: 0.061769s
25 Route time: 0.023148s
26 Total time: 0.110694s
27
28 Memory used: 584.000000 kB
```

Listing 4: Benchmark 4 - Log

4.17 Benchmark 5 - Placement Steps

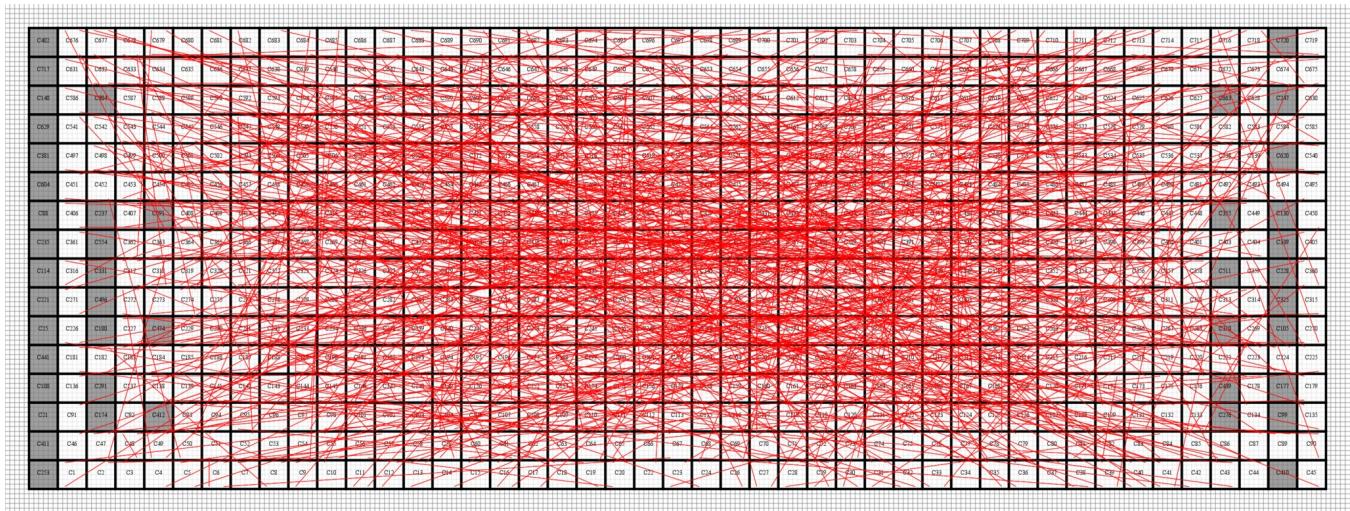


Figure 60: Benchmark 5 - Step 1/7 - Initial placement

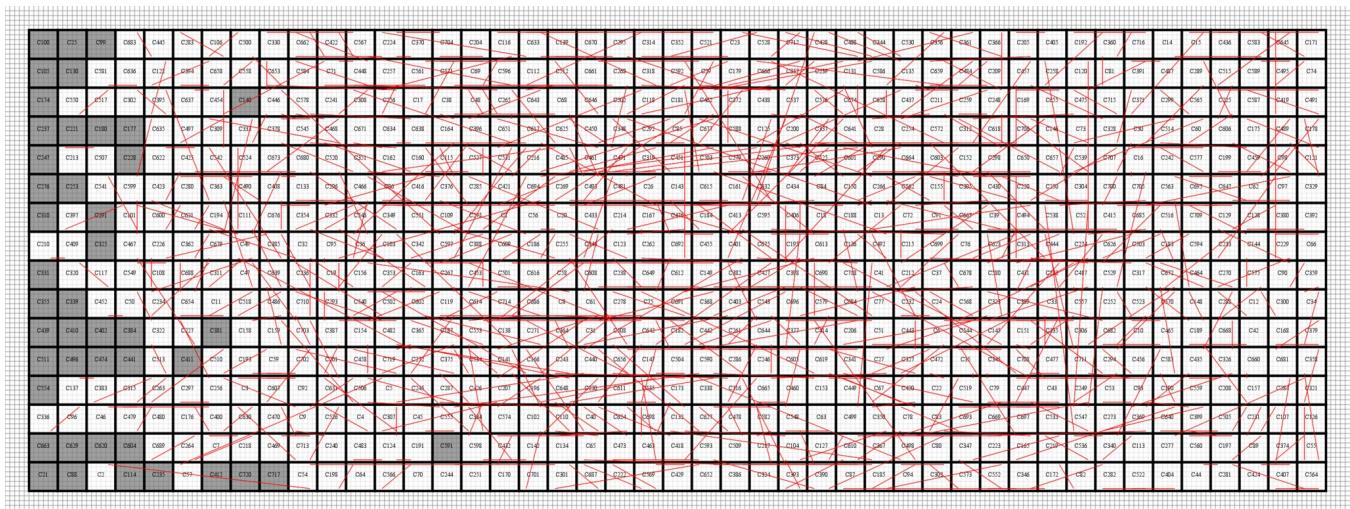


Figure 61: Benchmark 5 - Step 2/7 - Force directed placement

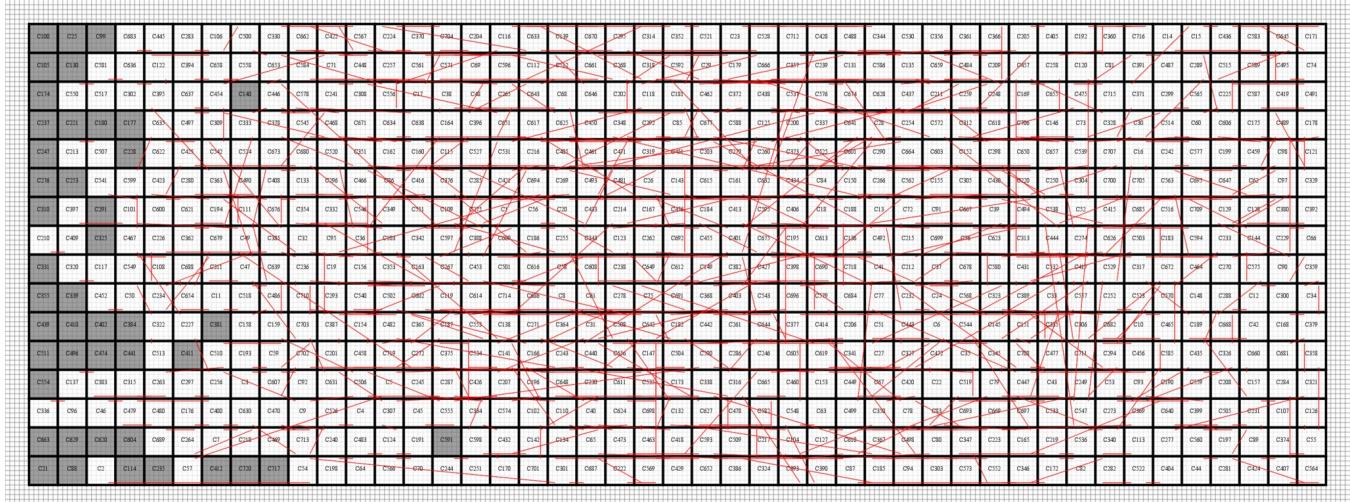


Figure 62: Benchmark 5 - Step 3/7 - Flip cells

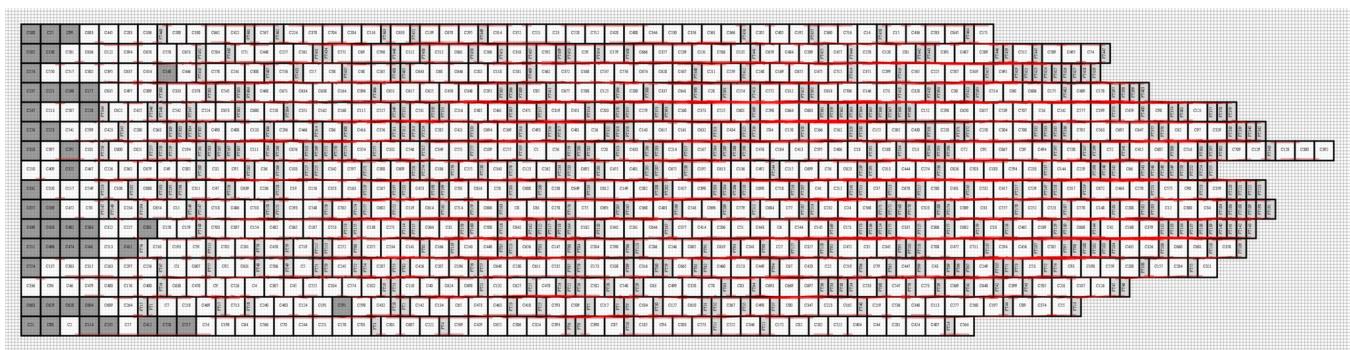


Figure 63: Benchmark 5 - Step 4/7 - Add feed-throughs

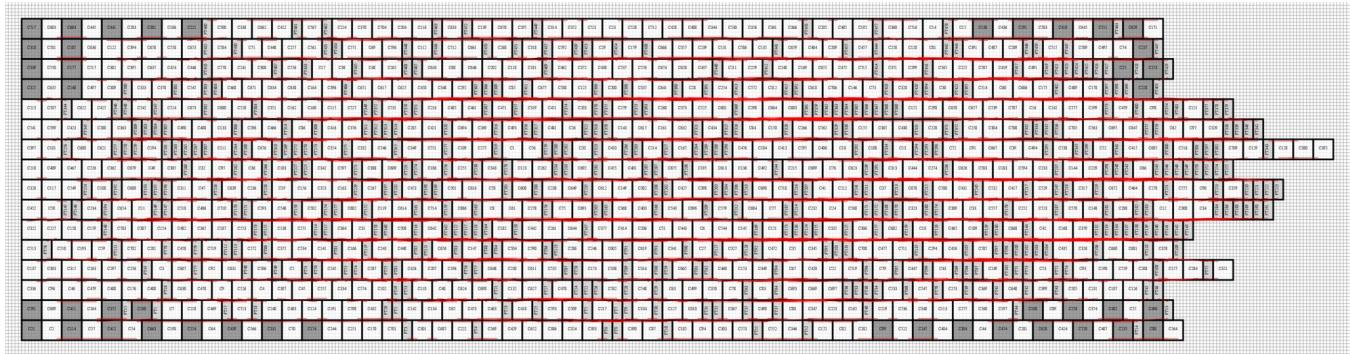


Figure 64: Benchmark 5 - Step 5/7 - Even out the row length

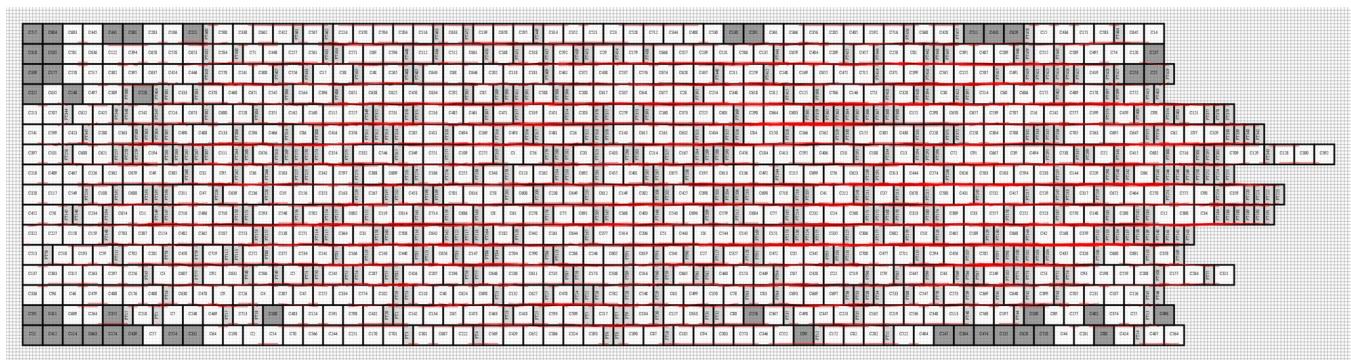


Figure 65: Benchmark 5 - Step 6/7 - Pull cells in the rows closer together

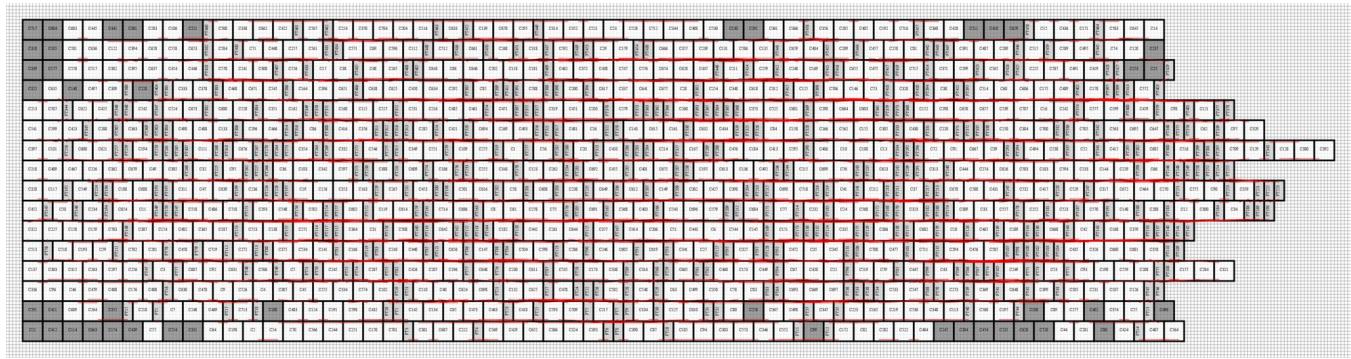


Figure 66: Benchmark 5 - Step 7/7 - Move feed-throughs to optimal location

4.18 Benchmark 5 - Final Routing

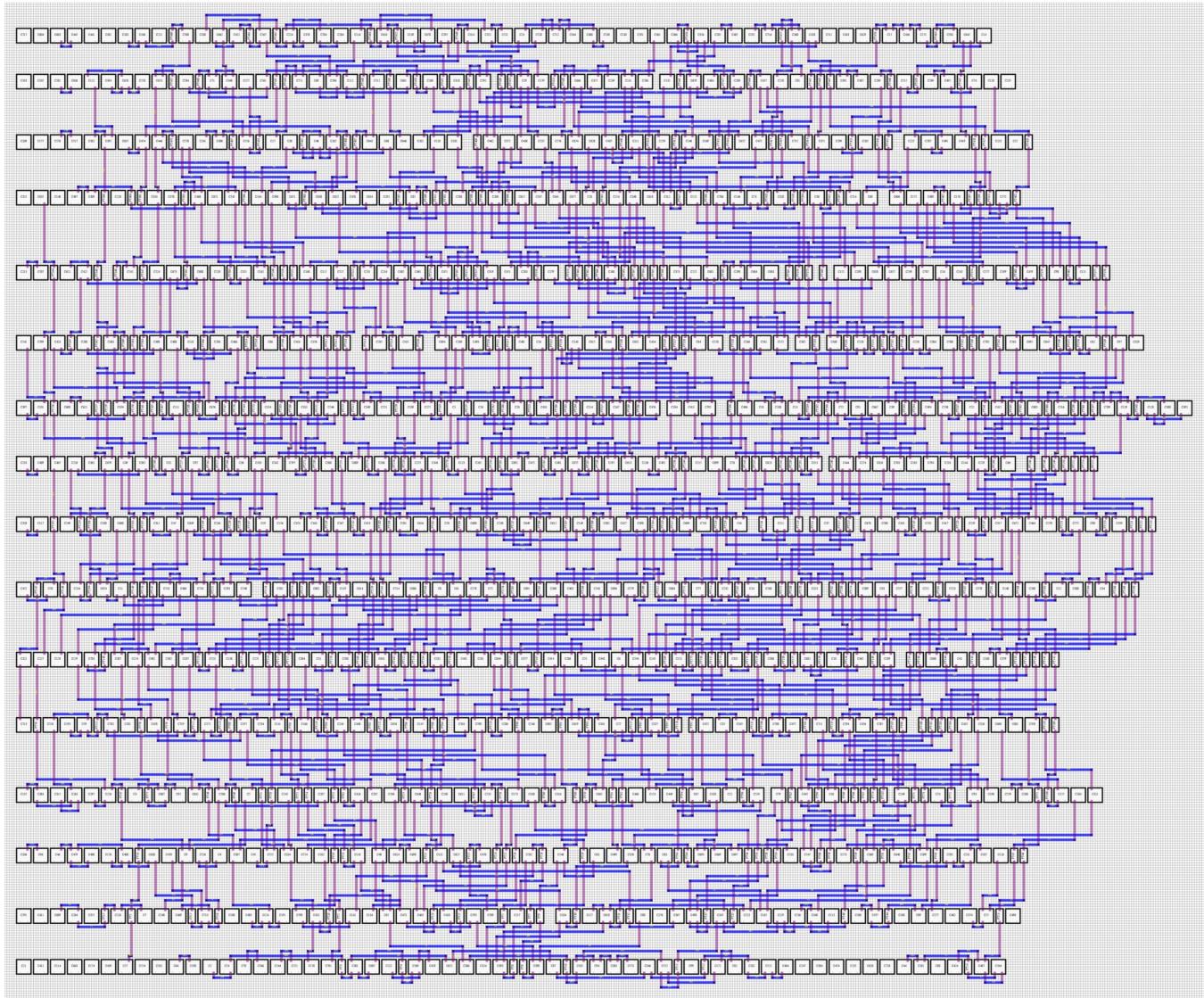


Figure 67: Benchmark 5 - Final routing

4.19 Benchmark 5 - Magic Screenshot

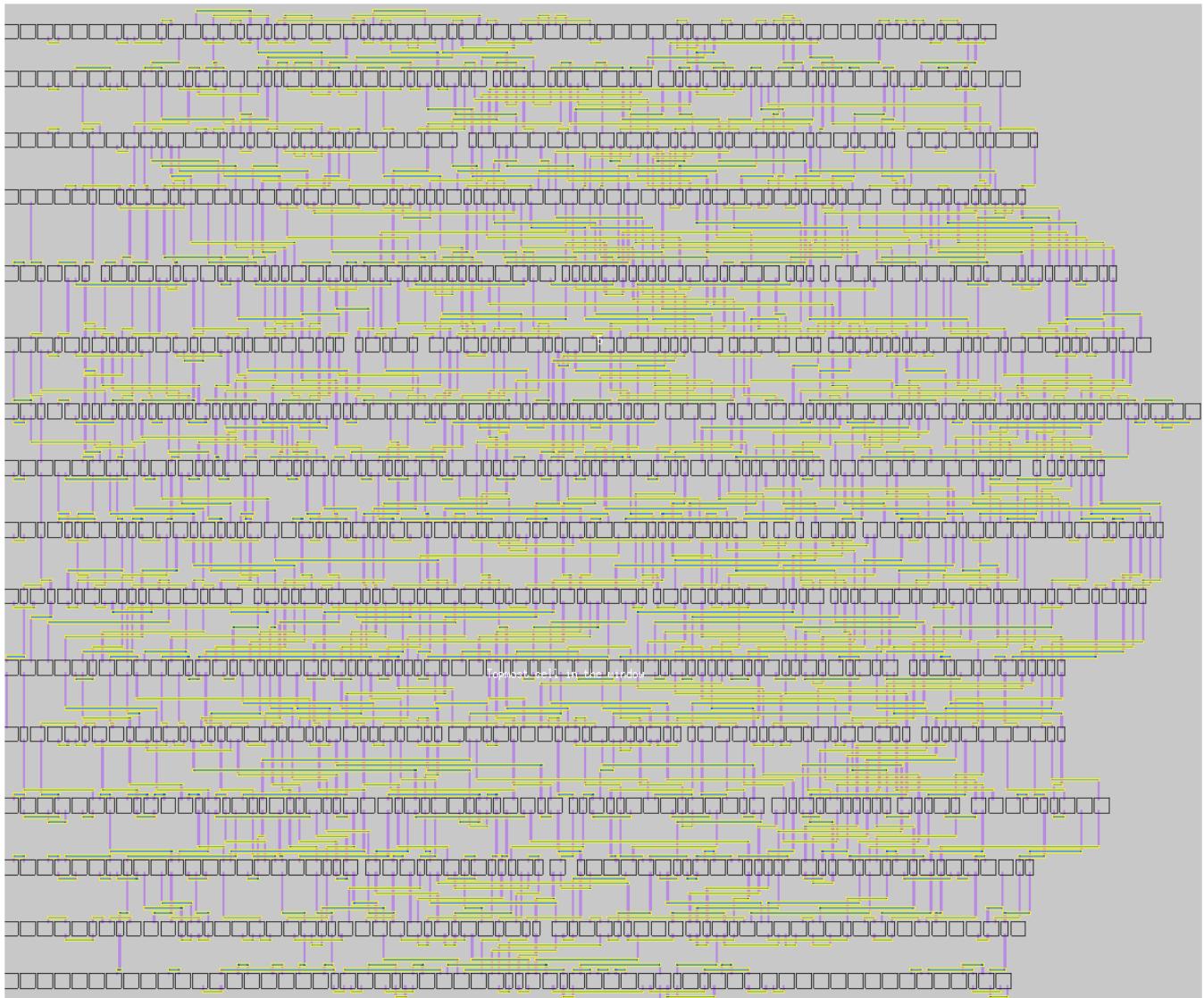


Figure 68: Benchmark 5- Magic Screenshot

4.20 Benchmark 5 - Log file

```
1 Number of cell: 720
2 Number of nets: 720
3
4 Placement
5 _____
6 Initial grid size: 45 16
7 Number of feed through cells: 464
8
9 Routing
10 _____
11 Number of vias: 2336
12 Total number of tracks: 146
13 Total wirelength: 41469
14
15 Size
16 _____
17 Total width: 486
18 Total height: 403
19 Total area: 195858
20 Squareness (width/height): 1.205955
21
22 Time
23 _____
24 Place time: 0.216536s
25 Route time: 0.083907s
26 Total time: 0.353846s
27
28 Memory used: 916.000000 kB
```

Listing 5: Benchmark 5 - Log

4.21 Benchmark 6 - Placement Steps

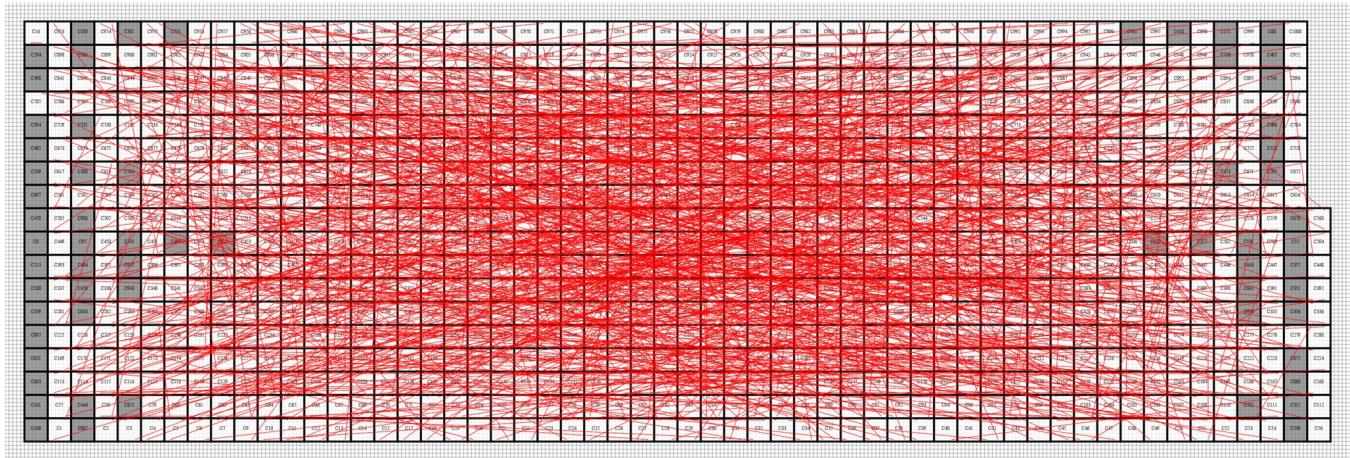


Figure 69: Benchmark 6 - Step 1/7 - Initial placement

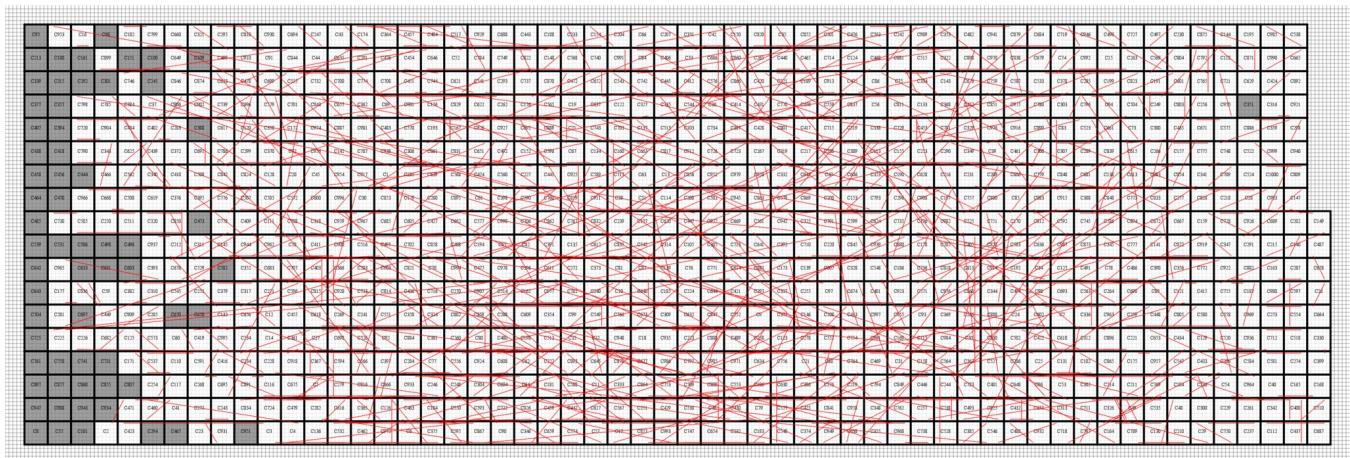


Figure 70: Benchmark 6 - Step 2/7 - Force directed placement

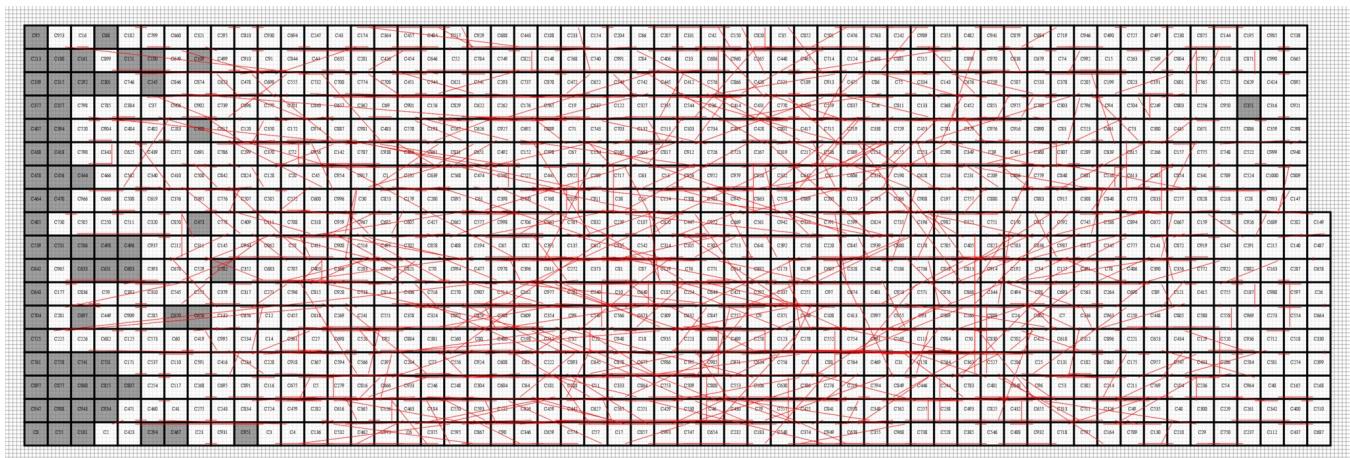


Figure 71: Benchmark 6 - Step 3/7 - Flip cells

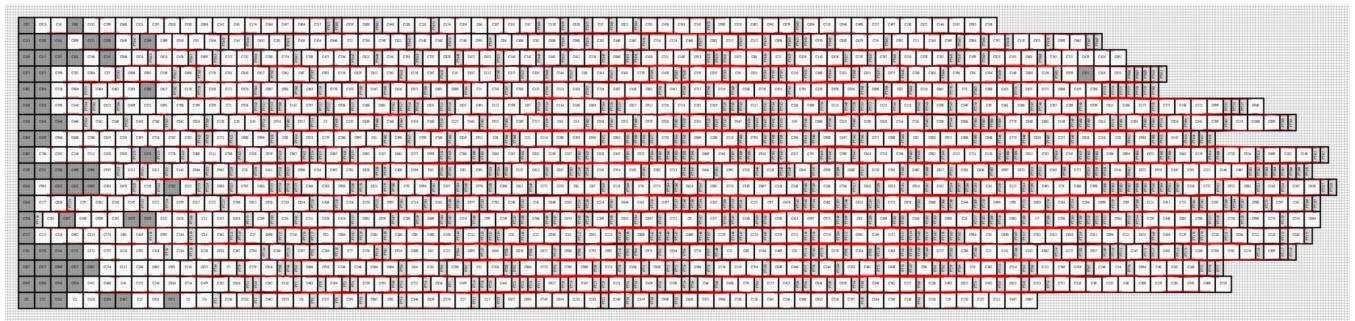


Figure 72: Benchmark 6 - Step 4/7 - Add feed-throughs

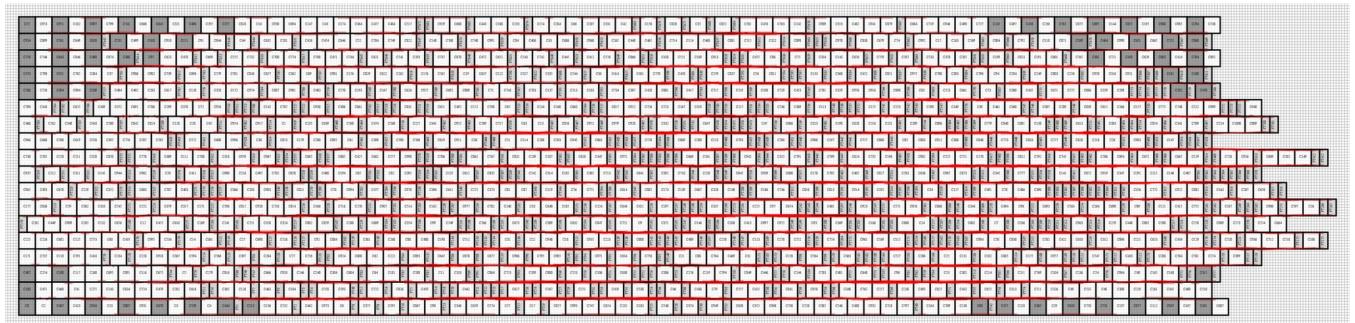


Figure 73: Benchmark 6 - Step 5/7 - Even out the row length

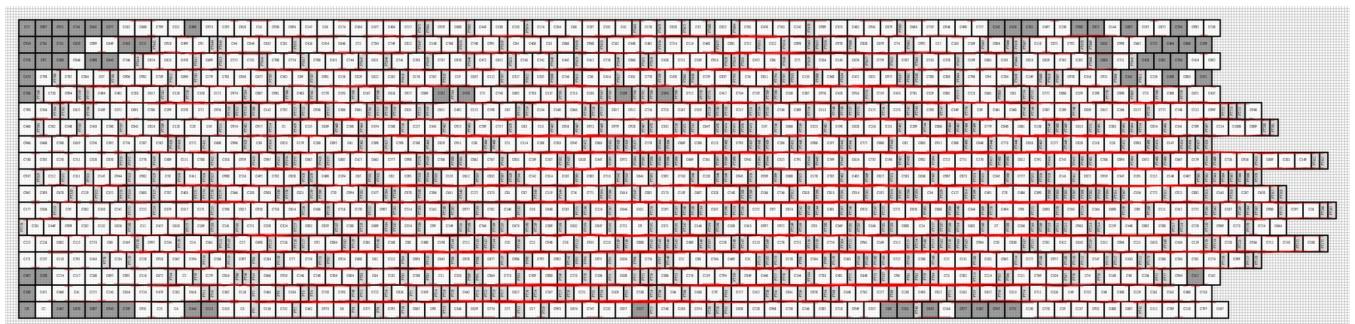


Figure 74: Benchmark 6 - Step 6/7 - Pull cells in the rows closer together

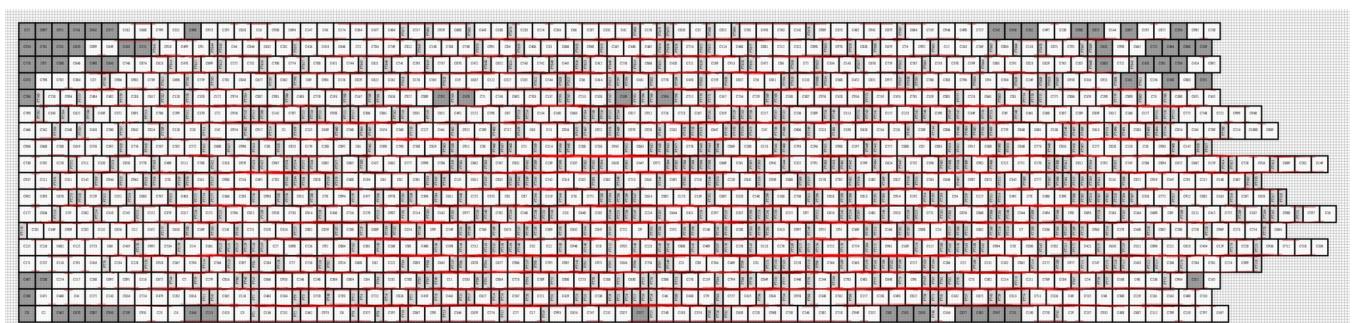


Figure 75: Benchmark 6 - Step 7/7 - Move feed-throughs to optimal location

4.22 Benchmark 6 - Final Routing

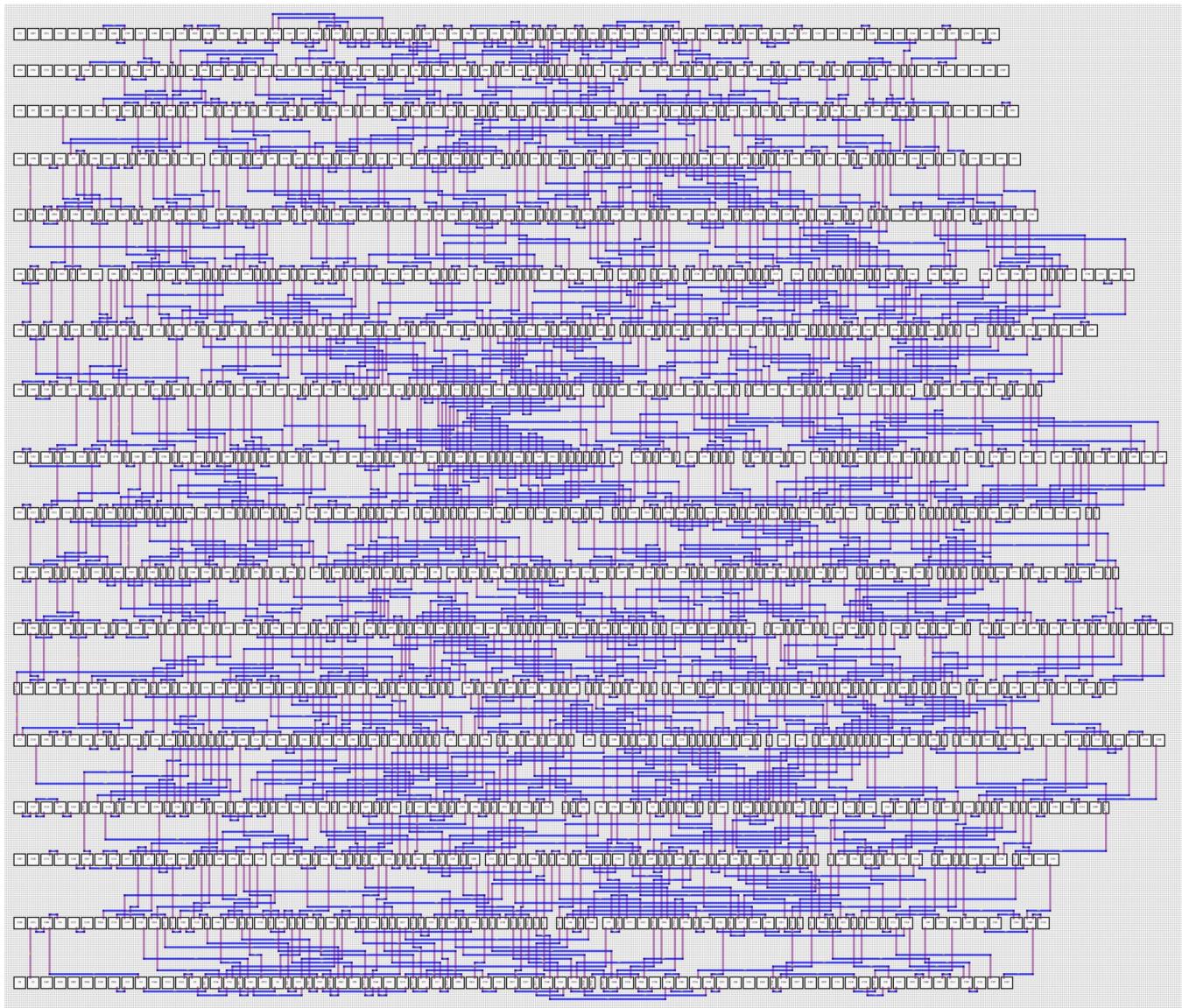


Figure 76: Benchmark 6 - Final routing

4.23 Benchmark 6 - Magic Screenshot

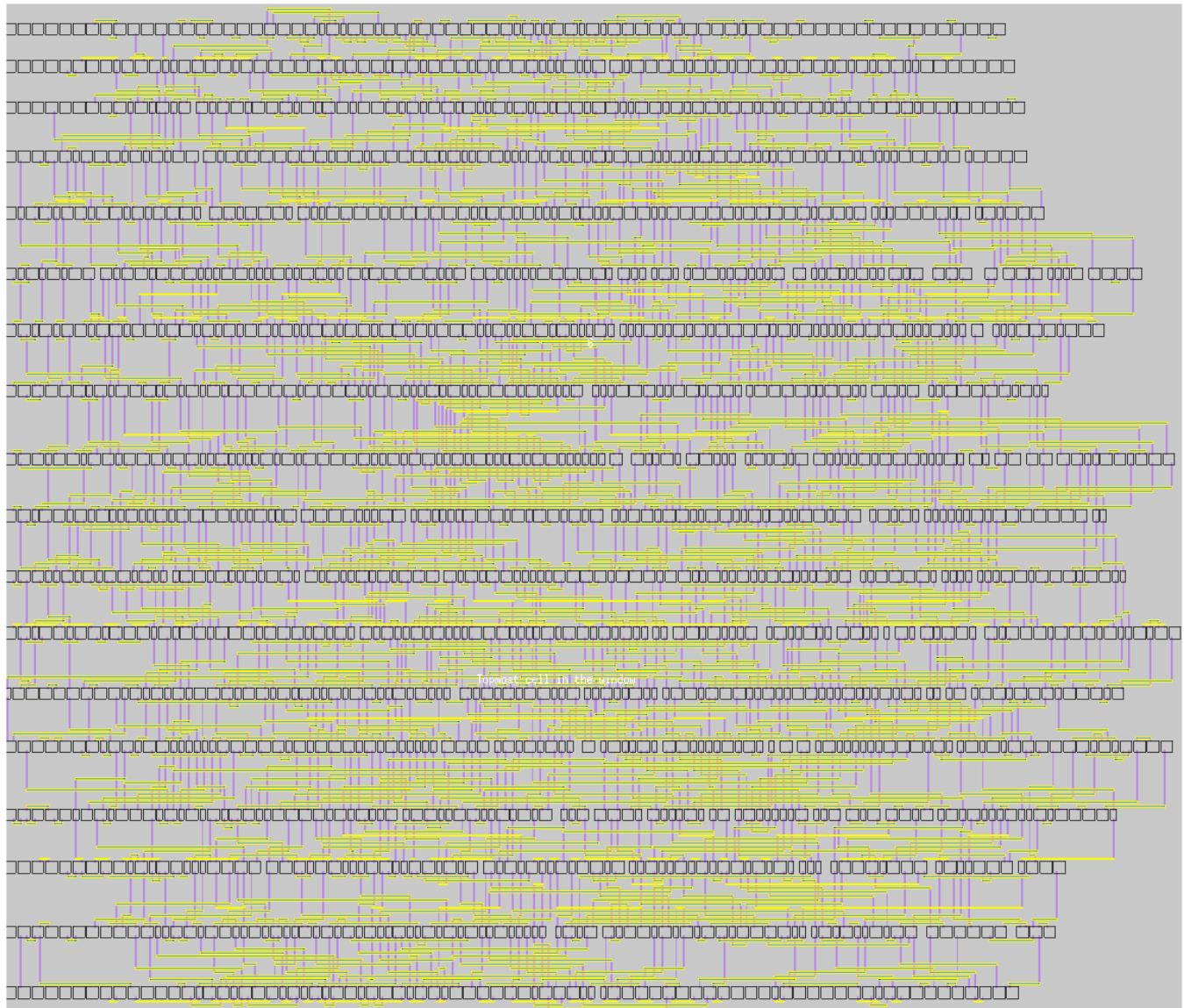


Figure 77: Benchmark 6- Magic Screenshot

4.24 Benchmark 6 - Log file

```
1 Number of cell: 1000
2 Number of nets: 1000
3
4 Placement
5 _____
6 Initial grid size: 56 18
7 Number of feed through cells: 685
8
9 Routing
10 _____
11 Number of vias: 3322
12 Total number of tracks: 194
13 Total wirelength: 70536
14
15 Size
16 _____
17 Total width: 602
18 Total height: 513
19 Total area: 308826
20 Squareness (width/height): 1.173489
21
22 Time
23 _____
24 Place time: 0.424019s
25 Route time: 0.168323s
26 Total time: 0.668167s
27
28 Memory used: 1208.000000 kB
```

Listing 6: Benchmark 6 - Log

4.25 Benchmark 7 - Placement Steps

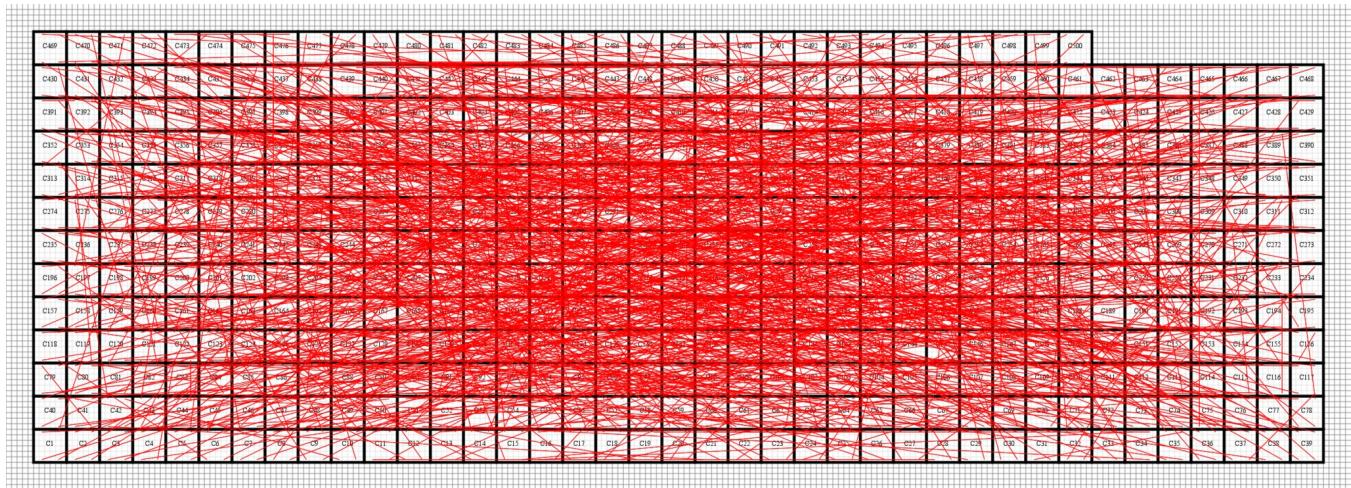


Figure 78: Benchmark 7 - Step 1/7 - Initial placement

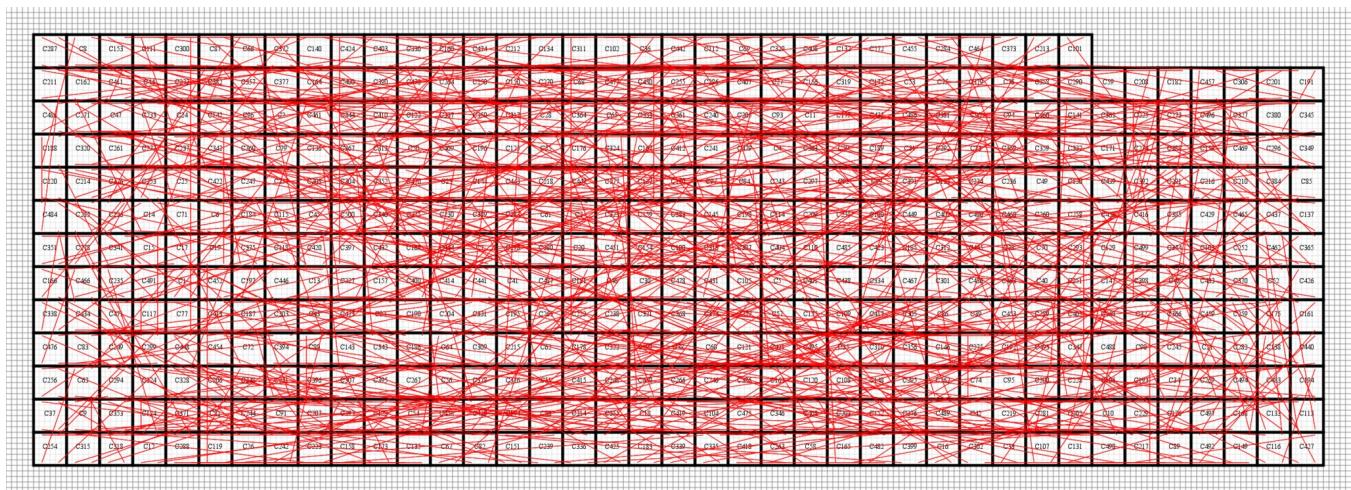


Figure 79: Benchmark 7 - Step 2/7 - Force directed placement

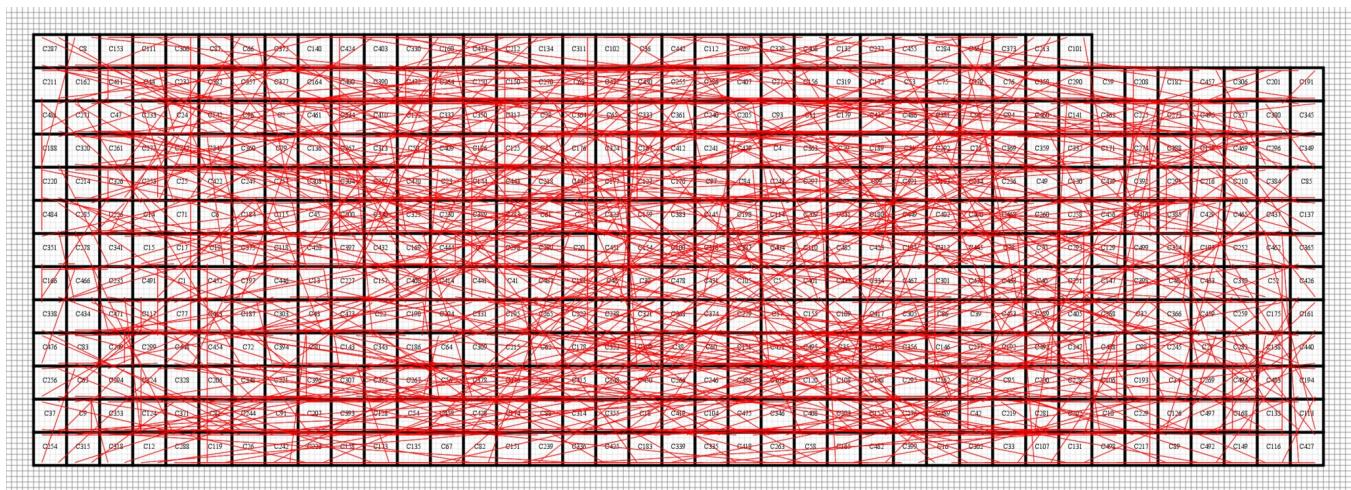


Figure 80: Benchmark 7 - Step 3/7 - Flip cells

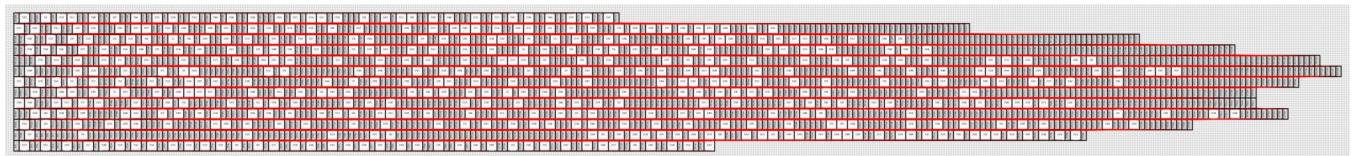


Figure 81: Benchmark 7 - Step 4/7 - Add feed-throughs

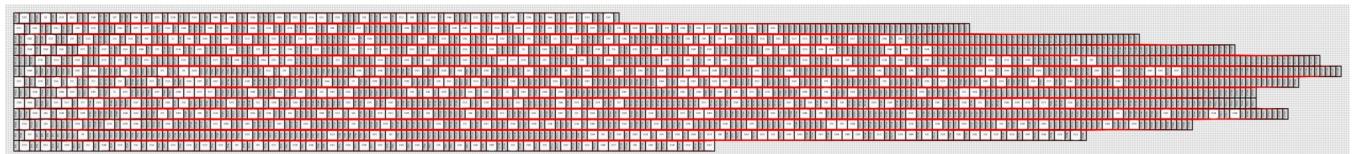


Figure 82: Benchmark 7 - Step 5/7 - Even out the row length

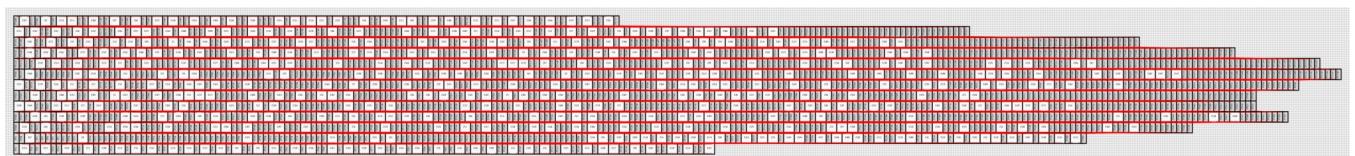


Figure 83: Benchmark 7 - Step 6/7 - Pull cells in the rows closer together

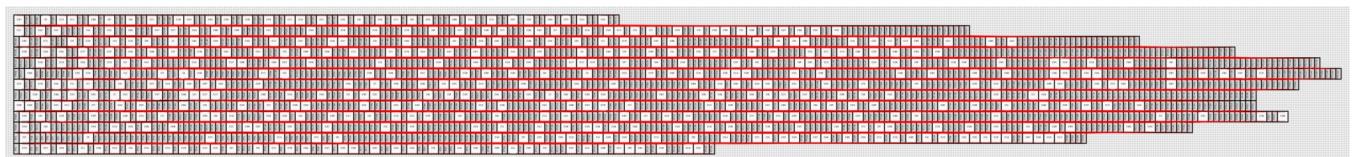


Figure 84: Benchmark 7 - Step 7/7 - Move feed-throughs to optimal location

4.26 Benchmark 7 - Final Routing

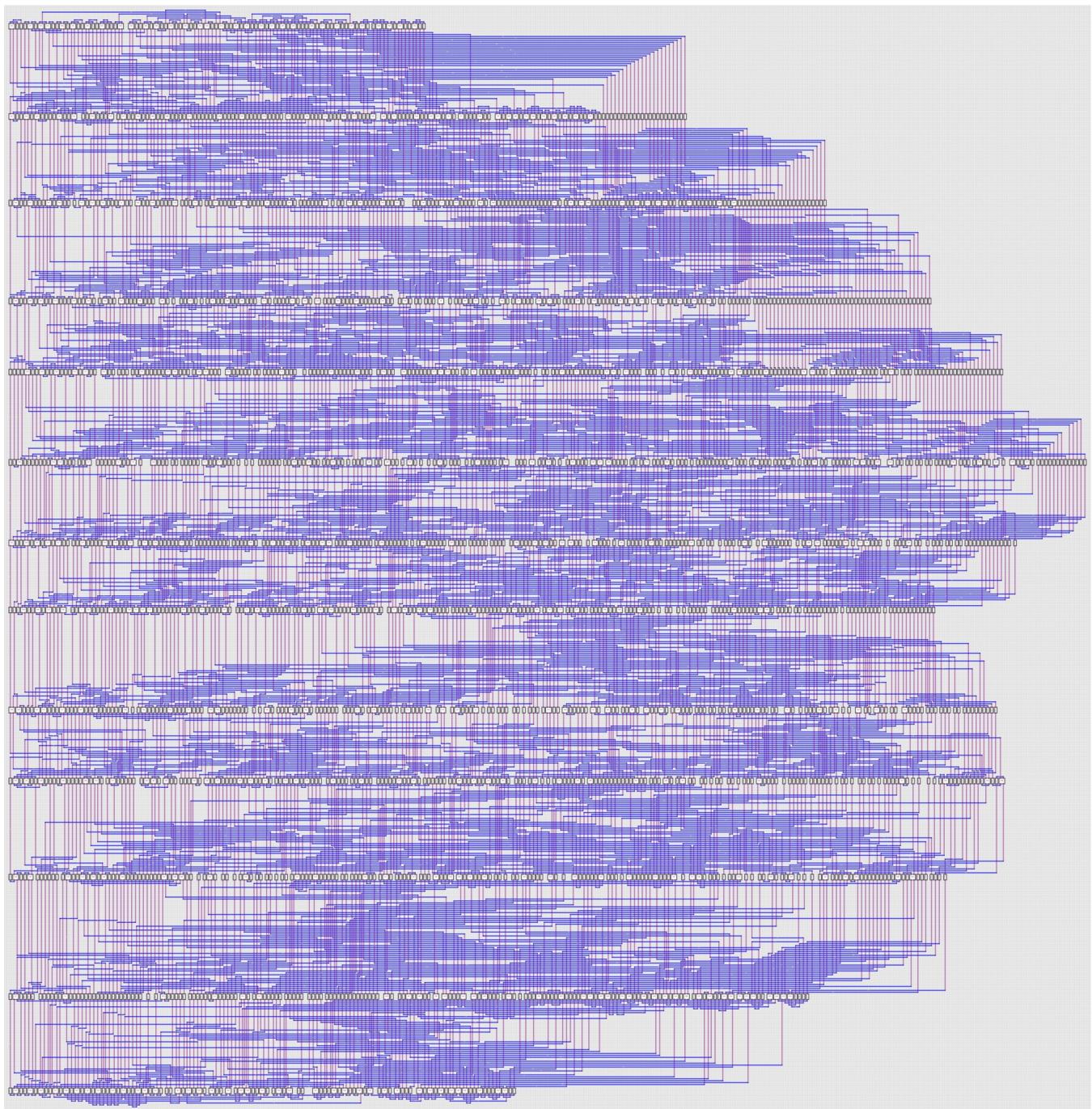


Figure 85: Benchmark 7 - Final routing

4.27 Benchmark 7 - Magic Screenshot

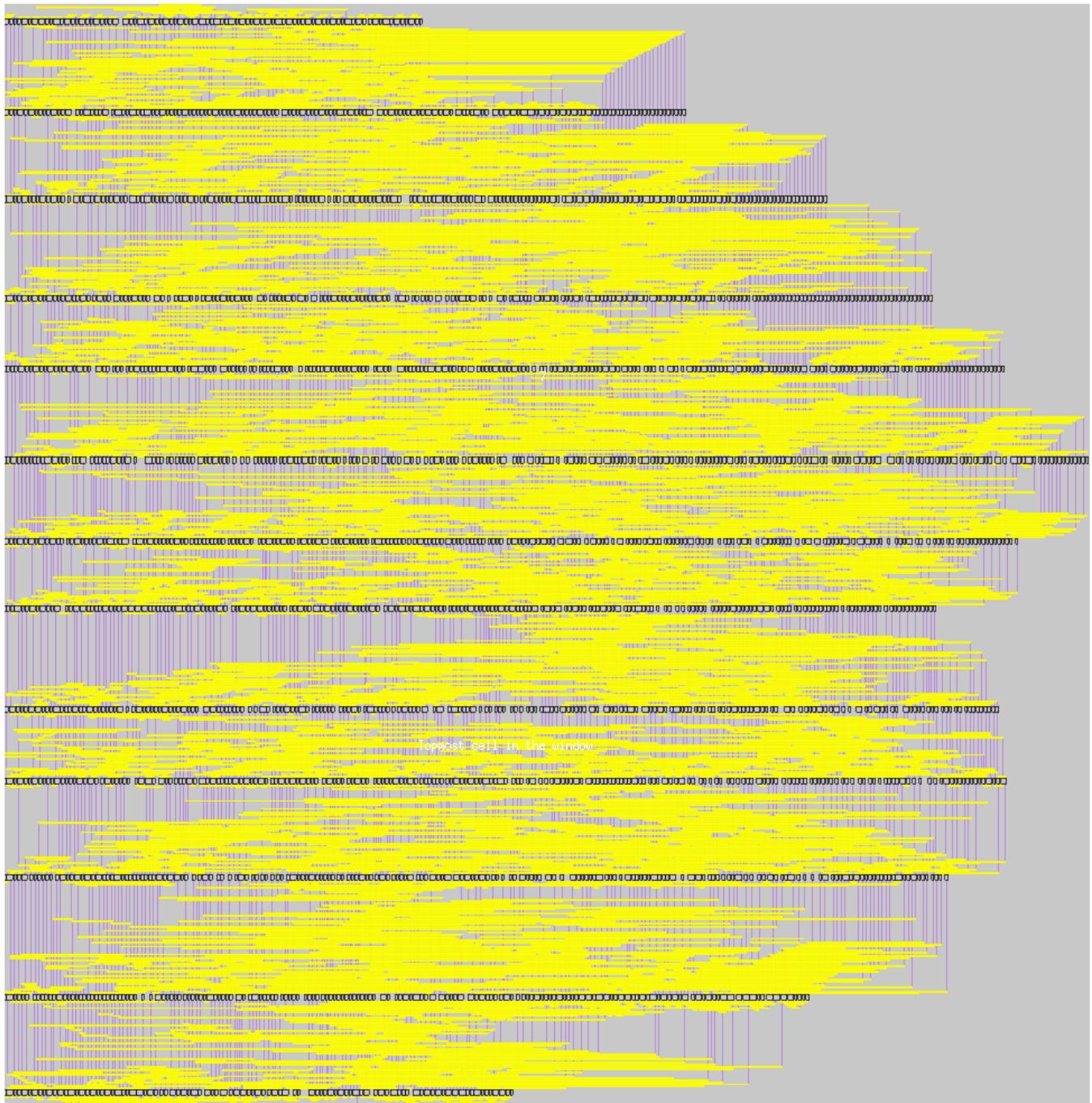


Figure 86: Benchmark 7 - Magic Screenshot

4.28 Benchmark 7 - Log file

```
1 Number of cell: 500
2 Number of nets: 1000
3
4 Placement
5 _____
6 Initial grid size: 39 13
7 Number of feed through cells: 1738
8
9 Routing
10 _____
11 Number of vias: 5448
12 Total number of tracks: 520
13 Total wirelength: 443767
14
15 Size
16 _____
17 Total width: 1110
18 Total height: 1130
19 Total area: 1254300
20 Squareness (width/height): 0.982301
21
22 Time
23 _____
24 Place time: 0.430179s
25 Route time: 1.023264s
26 Total time: 1.569695s
27
28 Memory used: 1512.000000 kB
```

Listing 7: Benchmark 7 - Log

4.29 Benchmark 8 - Placement Steps

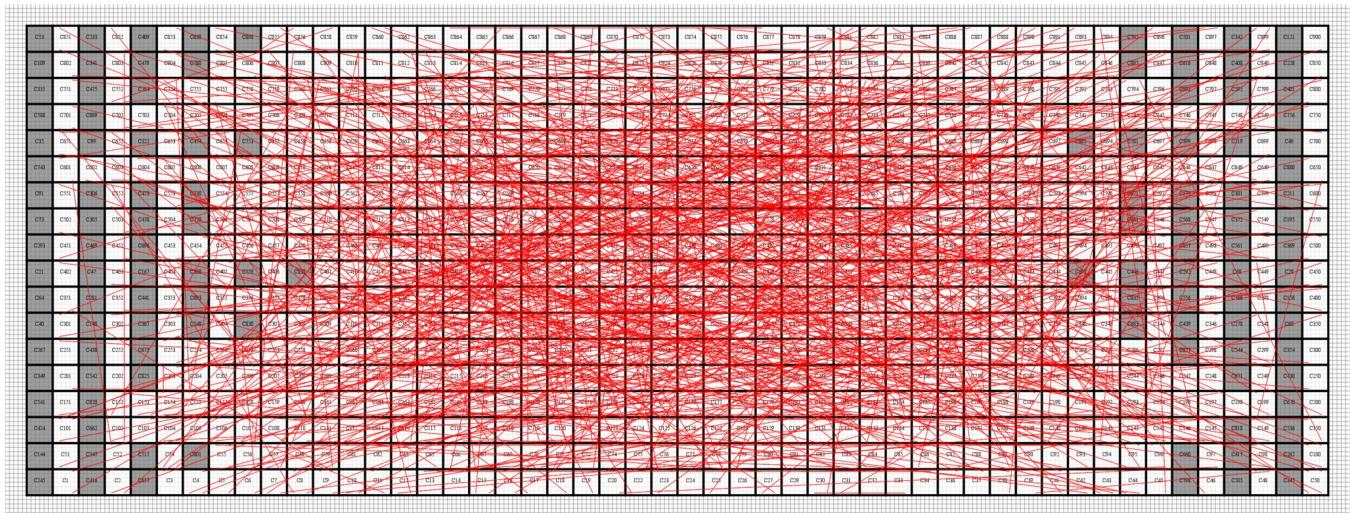


Figure 87: Benchmark 8 - Step 1/7 - Initial placement

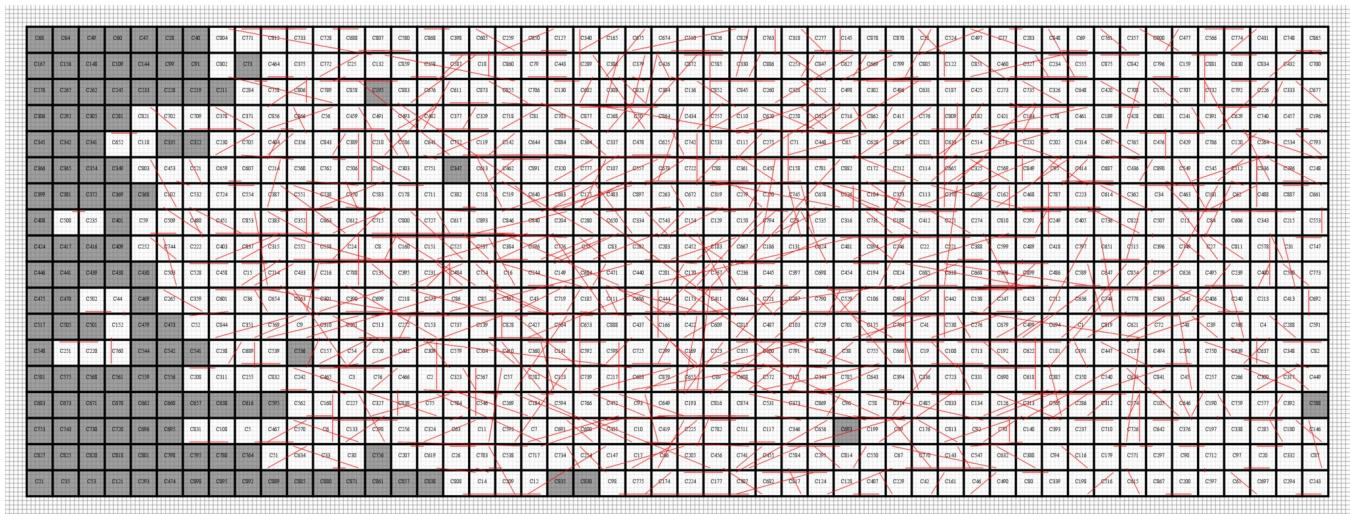


Figure 88: Benchmark 8 - Step 2/7 - Force directed placement

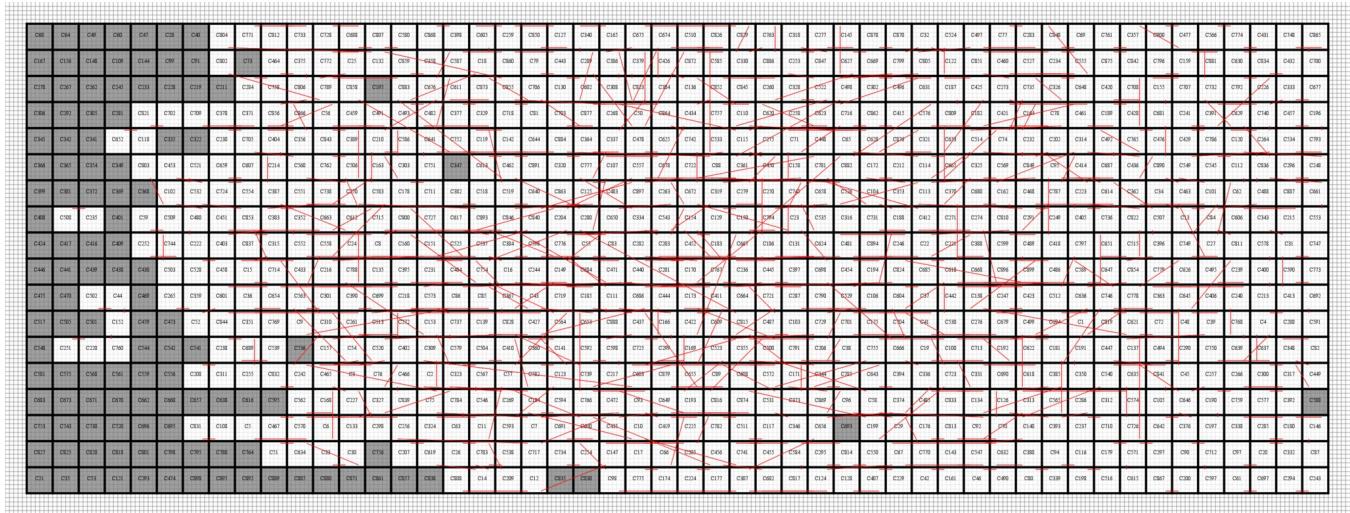


Figure 89: Benchmark 8- Step 3/7 - Flip cells

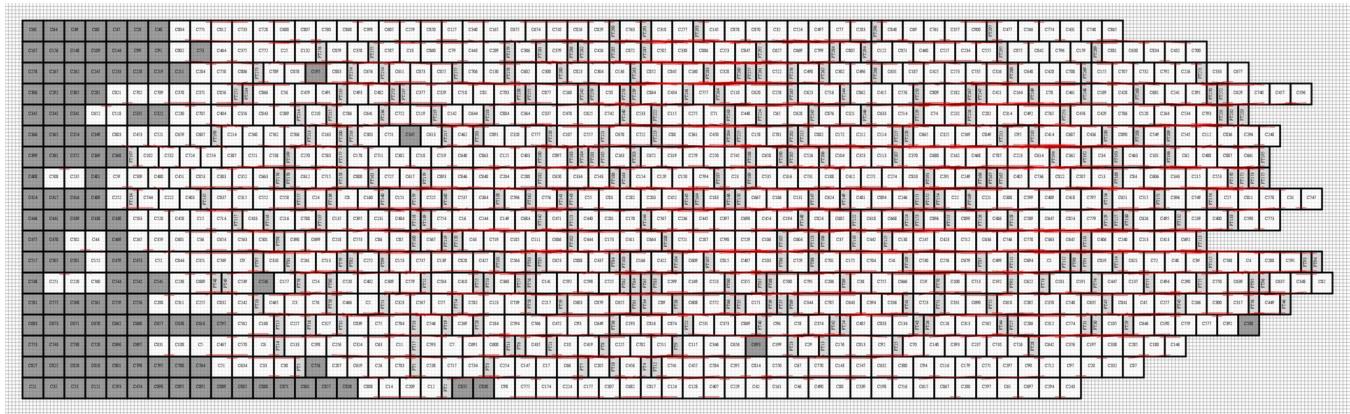


Figure 90: Benchmark 8- Step 4/7 - Add feed-throughs

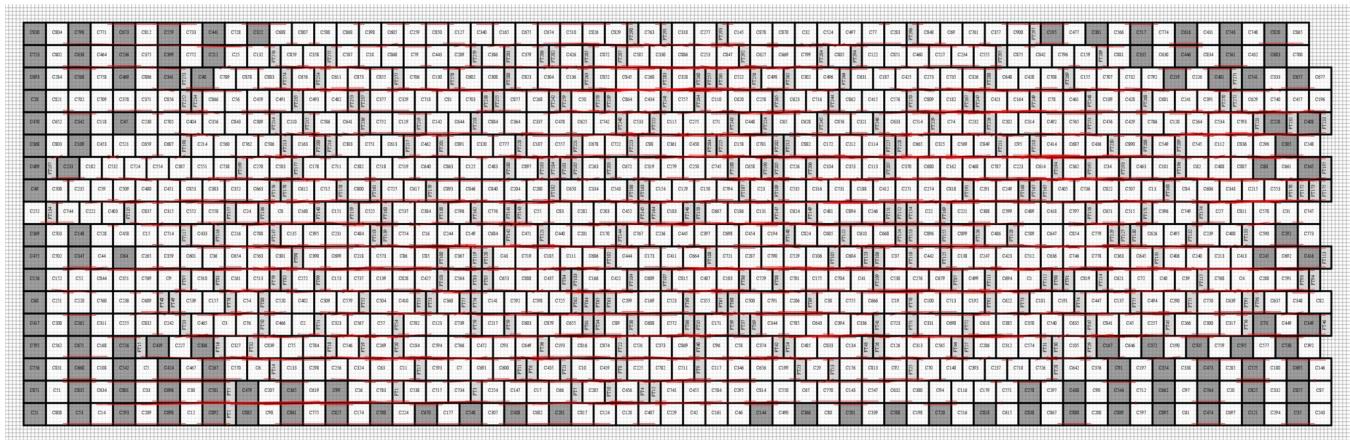
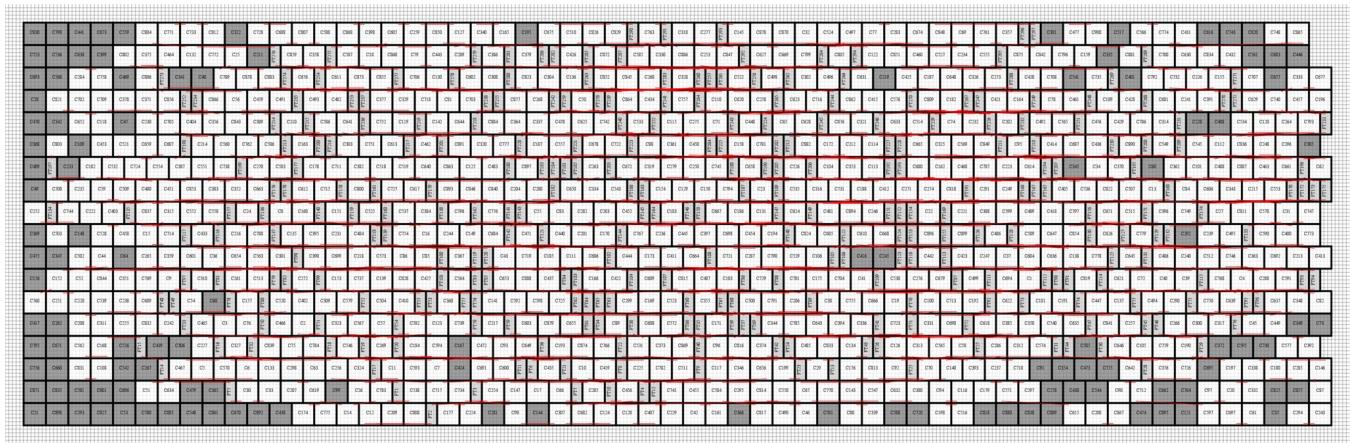
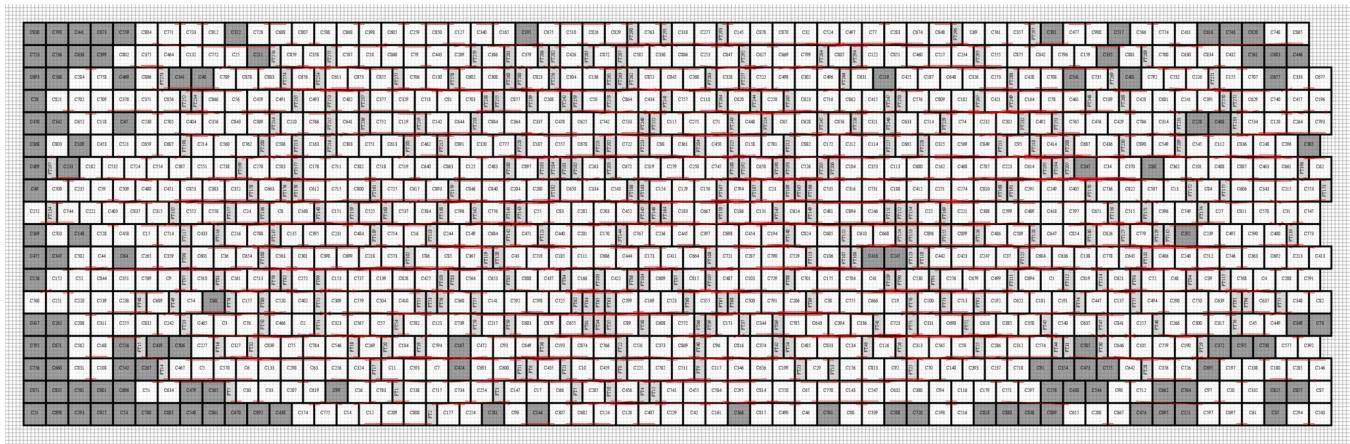


Figure 91: Benchmark 8- Step 5/7 - Even out the row length

**Figure 92:** Benchmark 8 - Step 6/7 - Pull cells in the rows closer together**Figure 93:** Benchmark 8 - Step 7/7 - Move feed-throughs to optimal location

4.30 Benchmark 8 - Final Routing

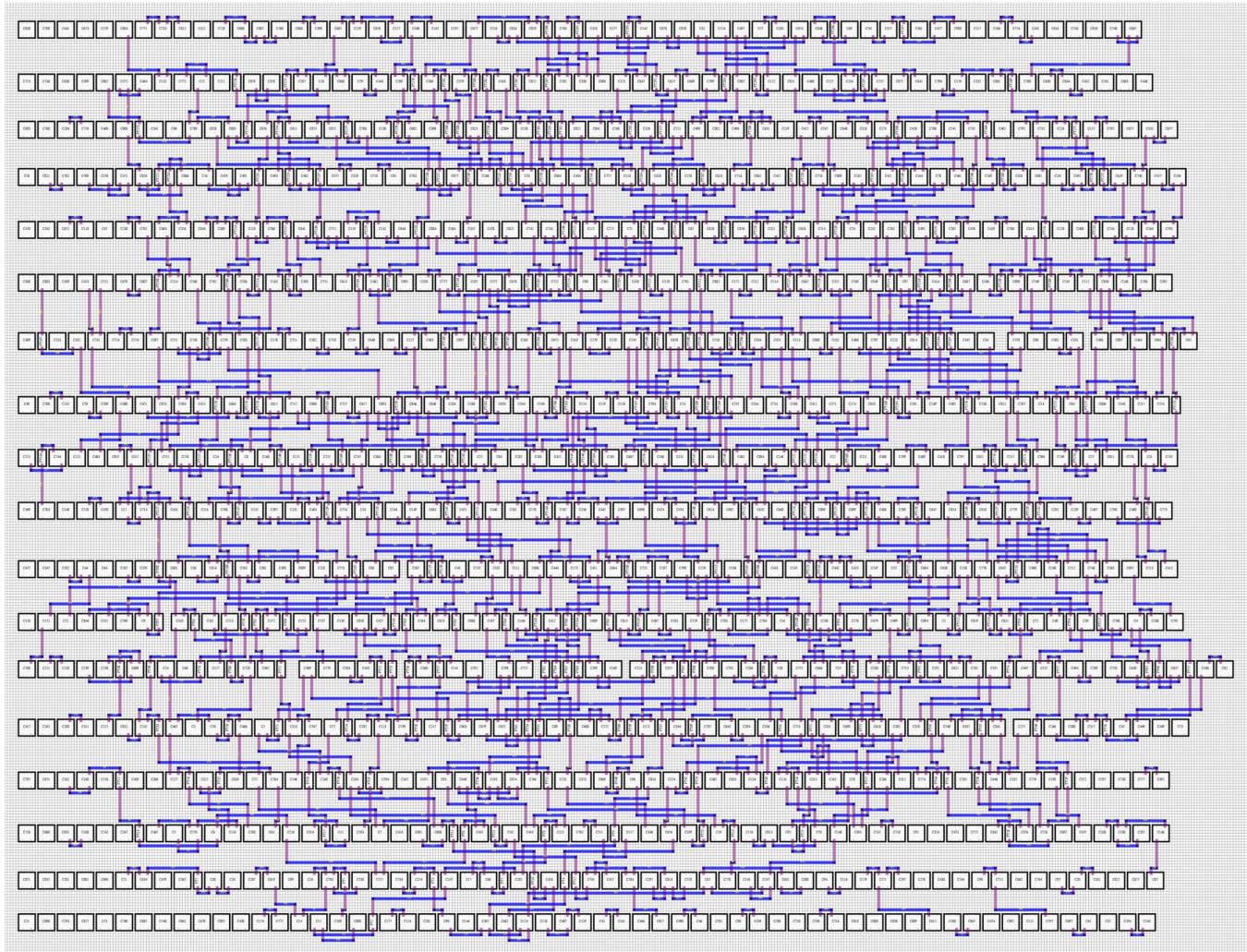


Figure 94: Benchmark 8 - Final routing

4.31 Benchmark 8 - Magic Screenshot

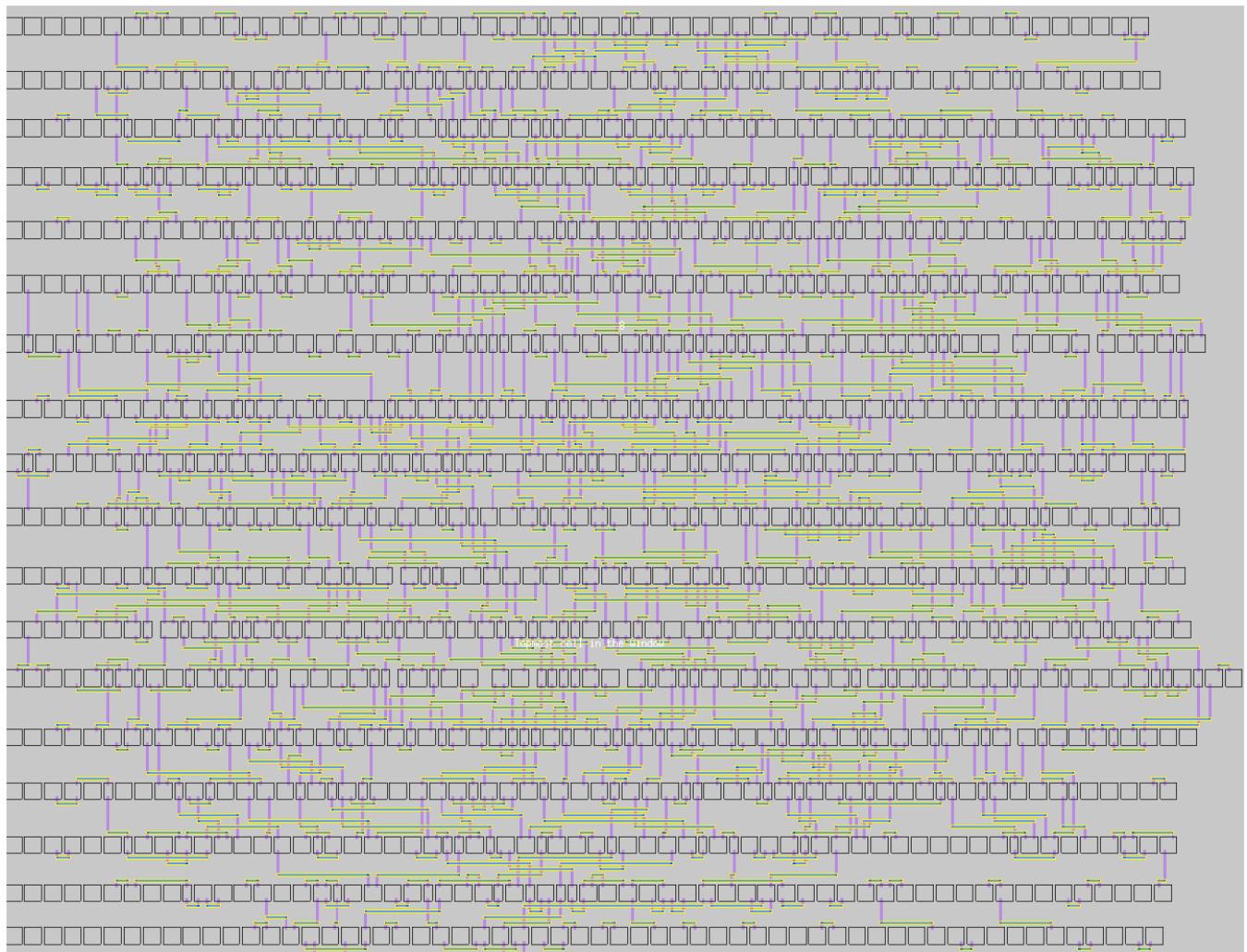


Figure 95: Benchmark 8- Magic Screenshot

4.32 Benchmark 8 - Log file

```
1 Number of cell: 900
2 Number of nets: 720
3
4 Placement
5 _____
6 Initial grid size: 50 18
7 Number of feed through cells: 297
8
9 Routing
10 _____
11 Number of vias: 1994
12 Total number of tracks: 104
13 Total wirelength: 22907
14
15 Size
16 _____
17 Total width: 437
18 Total height: 333
19 Total area: 145521
20 Squareness (width/height): 1.312312
21
22 Time
23 _____
24 Place time: 0.238644s
25 Route time: 0.051257s
26 Total time: 0.339161s
27
28 Memory used: 916.000000 kB
```

Listing 8: Benchmark 8 - Log

4.33 Benchmark 9 - Placement Steps

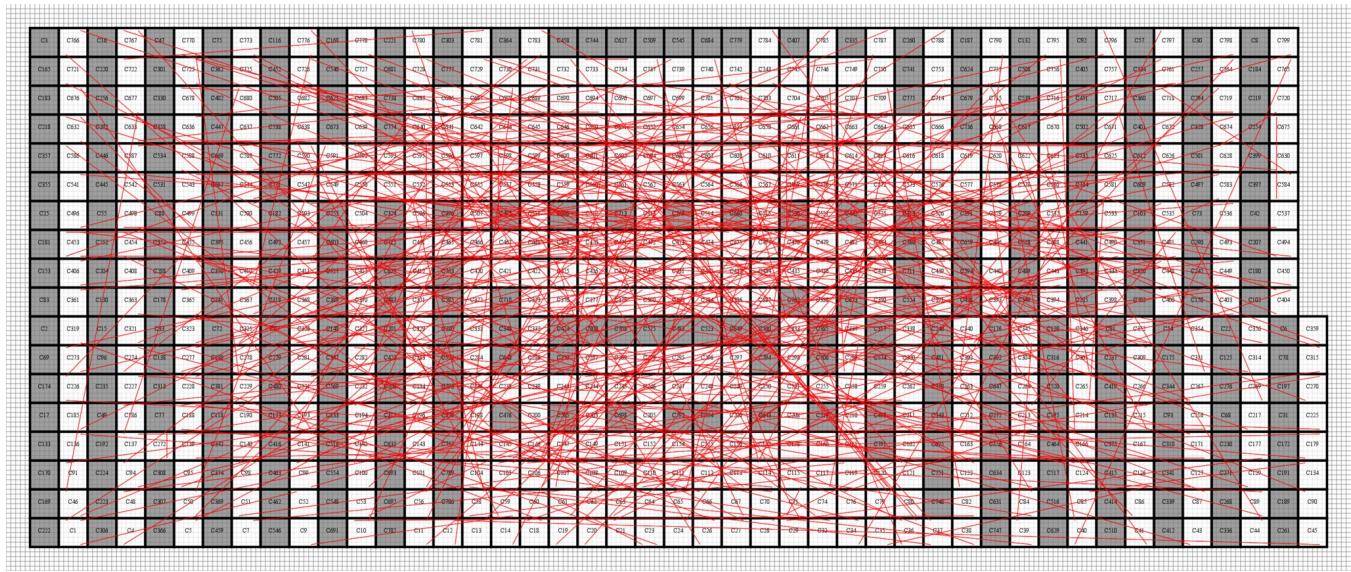


Figure 96: Benchmark 9 - Step 1/7 - Initial placement

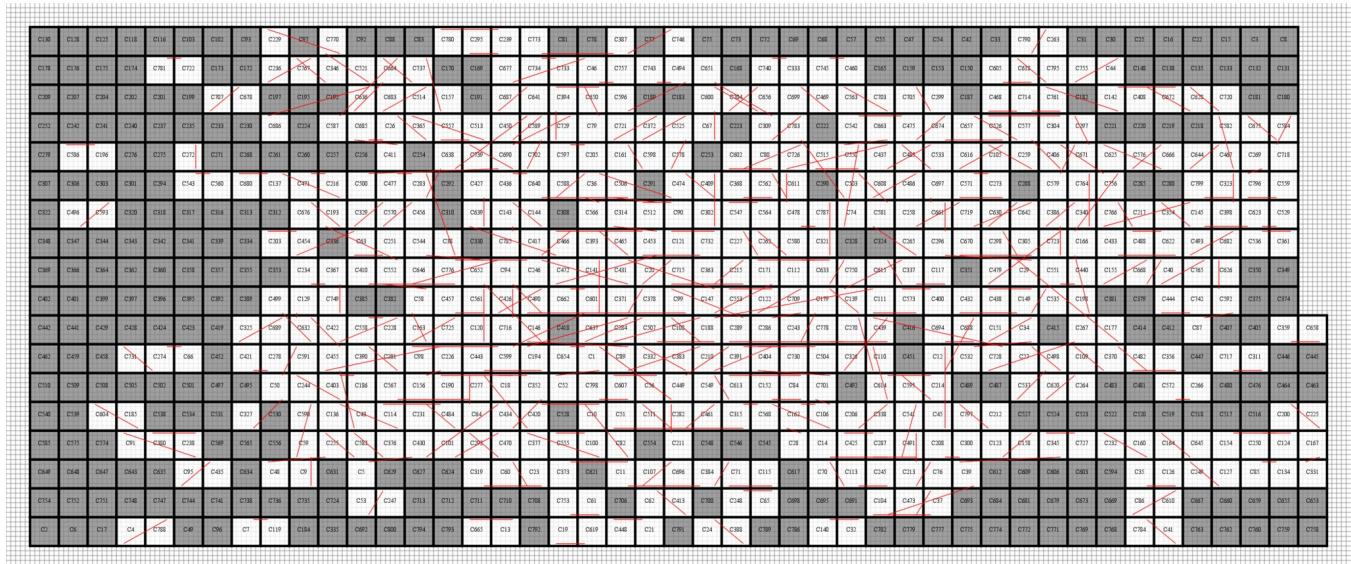


Figure 97: Benchmark 9 - Step 2/7 - Force directed placement

Figure 98: Benchmark 9- Step 3/7 - Flip cells

Figure 99: Benchmark 9- Step 4/7 - Add feed-throughs

Figure 100: Benchmark 9 - Step 5/7 - Even out the row length

Figure 101: Benchmark 9 - Step 6/7 - Pull cells in the rows closer together

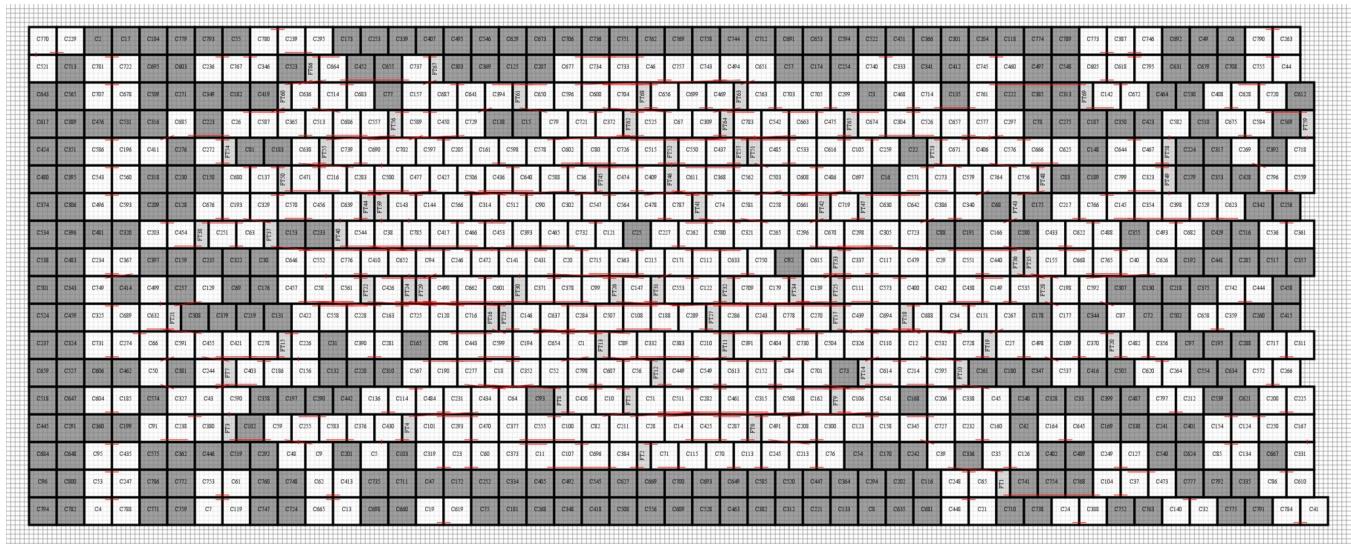


Figure 102: Benchmark 9 - Step 7/7 - Move feed-throughs to optimal location

4.34 Benchmark 9 - Final Routing

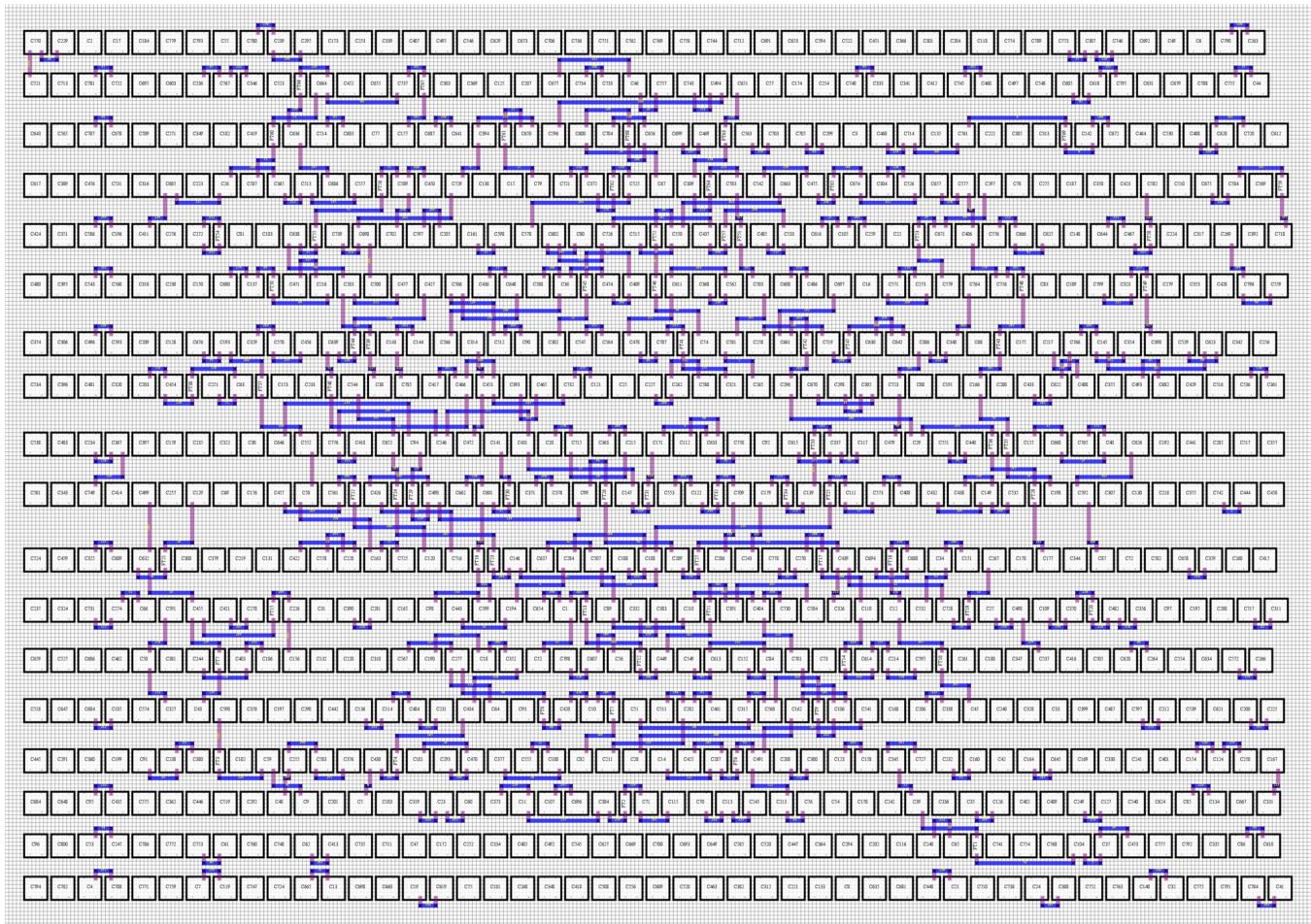


Figure 103: Benchmark 9 - Final routing

4.35 Benchmark 9 - Magic Screenshot



Figure 104: Benchmark 9 - Magic Screenshot

4.36 Benchmark 9 - Log file

```
1 Number of cell: 800
2 Number of nets: 360
3
4 Placement
5 _____
6 Initial grid size: 45 18
7 Number of feed through cells: 69
8
9 Routing
10 _____
11 Number of vias: 842
12 Total number of tracks: 52
13 Total wirelength: 6097
14
15 Size
16 _____
17 Total width: 328
18 Total height: 229
19 Total area: 75112
20 Squareness (width/height): 1.432314
21
22 Time
23 _____
24 Place time: 0.165468s
25 Route time: 0.011609s
26 Total time: 0.204302s
27
28 Memory used: 668.000000 kB
```

Listing 9: Benchmark 9 - Log

4.37 Benchmark 10 - Placement Steps

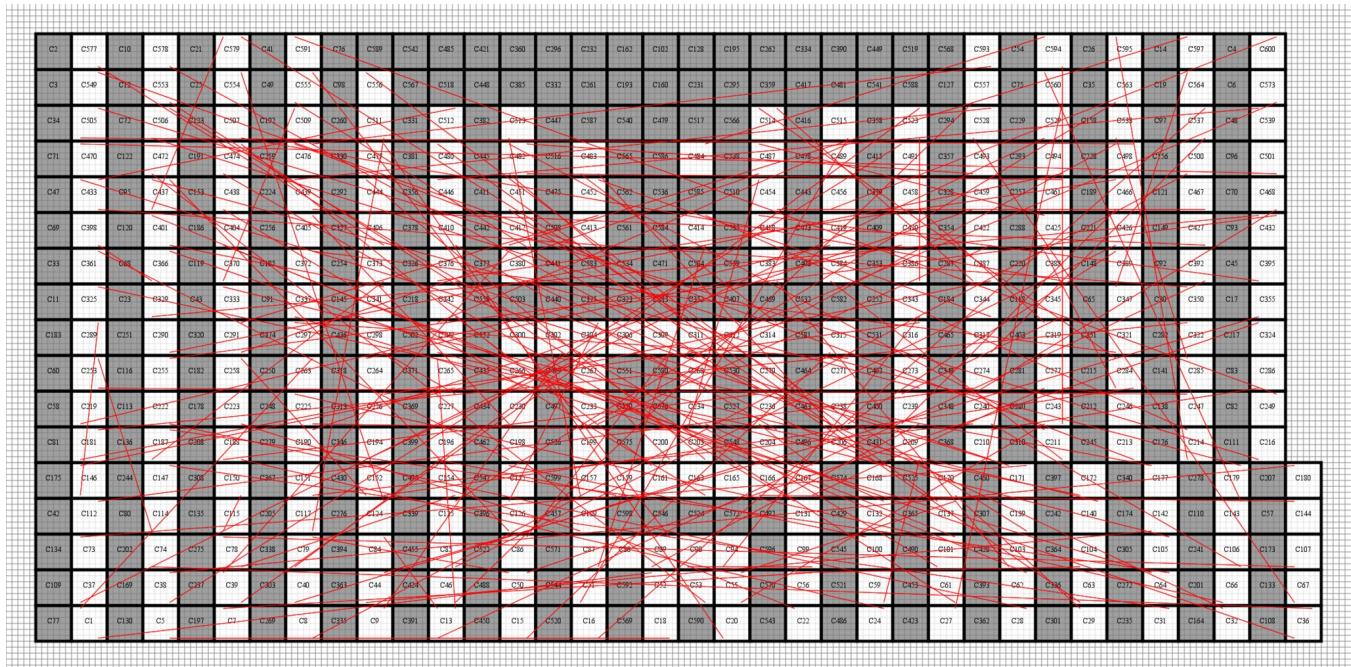


Figure 105: Benchmark 10 - Step 1/7 - Initial placement

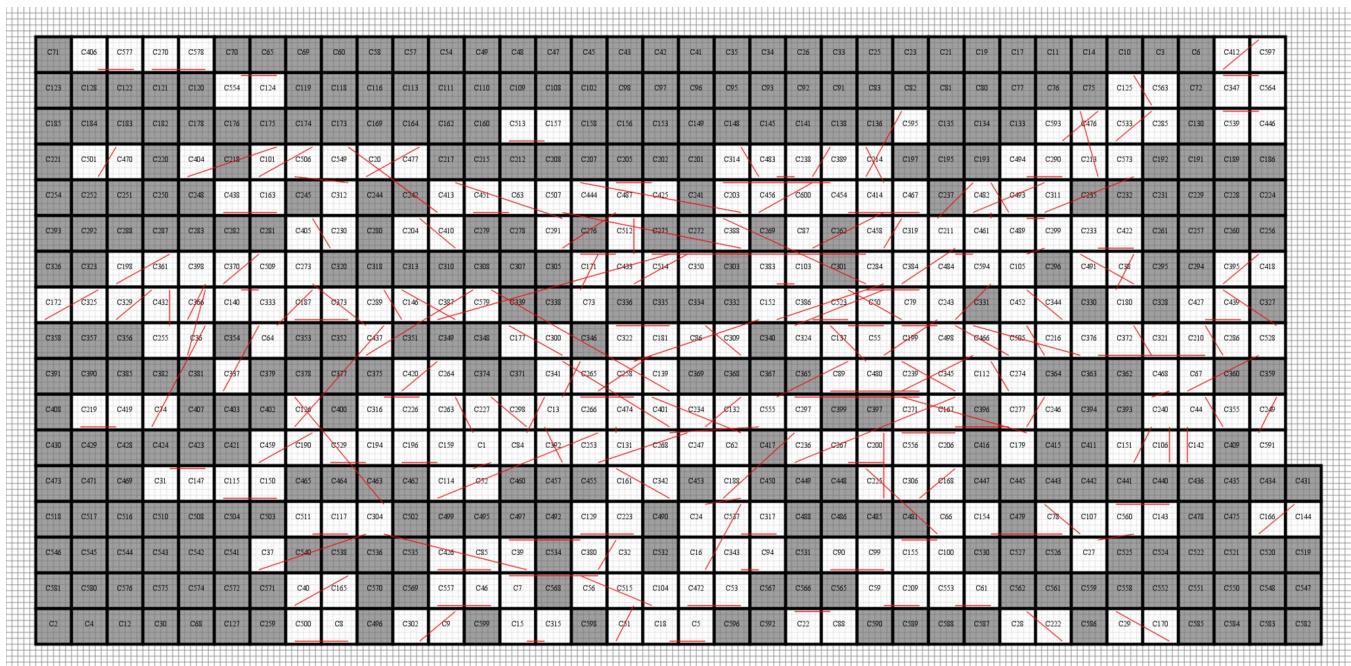


Figure 106: Benchmark 10 - Step 2/7 - Force directed placement

C71	C406	C377	C270	C578	C701	C65	C68	C60	C53	C77	C54	C48	C47	C45	C48	C42	C46	C35	C34	C36	C33	C25	C23	C21	C19	C17	C11	C14	C10	C9	C6	C412	C97		
C123	C138	C122	C121	C100	C554	C124	C119	C118	C116	C113	C111	C110	C109	C108	C102	C98	C97	C96	C95	C93	C92	C91	C83	C82	C81	C80	C77	C76	C75	C125	C63	C72	C347	C564	
C185	C184	C183	C182	C178	C176	C175	C174	C173	C169	C164	C162	C160	C513	C151	C150	C153	C149	C141	C141	C138	C136	C95	C135	C134	C131	C131	C93	C406	C333	C285	C130	C59	C446		
C221	C381	C470	C238	C404	C218	C101	C266	C548	C20	C477	C217	C215	C212	C209	C207	C203	C202	C201	C314	C483	C338	C389	C214	C197	C195	C191	C484	C390	C213	C573	C192	C191	C189	C186	
C254	C251	C151	C150	C248	C488	C163	C245	C162	C244	C240	C413	C401	C53	C501	C444	C467	C415	C241	C303	C456	C400	C414	C414	C467	C231	C62	C463	C311	C232	C231	C229	C228	C224		
C293	C392	C388	C387	C381	C381	C485	C330	C304	C410	C279	C278	C291	C216	C212	C235	C272	C388	C389	C387	C362	C435	C319	C211	C461	C489	C399	C331	C422	C361	C257	C360	C236			
C315	C313	C198	C361	C398	C710	C309	C273	C330	C313	C311	C310	C308	C307	C303	C301	C314	C150	C303	C381	C164	C301	C344	C394	C105	C306	C481	C395	C393	C418	C313	C421	C489	C327		
C172	C311	C129	C412	C366	C140	C333	C187	C373	C289	C146	C387	C179	C339	C338	C73	C336	C335	C334	C332	C152	C386	C339	C339	C179	C414	C381	C412	C344	C330	C180	C332	C421	C489	C327	
C318	C317	C357	C356	C353	C354	C364	C353	C351	C407	C351	C349	C348	C177	C286	C345	C322	C181	C384	C309	C340	C334	C171	C155	C199	C486	C366	C362	C216	C316	C372	C311	C210	C366	C328	
C391	C380	C381	C382	C381	C381	C379	C378	C379	C375	C375	C403	C364	C314	C371	C341	C365	C238	C139	C367	C365	C395	C480	C239	C345	C345	C364	C381	C380	C486	C367	C360	C359	C349		
C408	C119	C419	C174	C407	C403	C402	C126	C400	C116	C126	C263	C227	C298	C113	C266	C414	C401	C334	C132	C355	C397	C399	C387	C711	C396	C377	C346	C394	C393	C340	C444	C315	C340		
C430	C429	C433	C424	C423	C421	C419	C199	C139	C194	C196	C159	C1	C34	C392	C213	C131	C268	C247	C62	C417	C336	C267	C300	C556	C306	C416	C178	C415	C111	C106	C142	C429	C391		
C473	C471	C469	C31	C147	C115	C130	C465	C464	C463	C463	C114	C52	C460	C417	C455	C161	C342	C433	C188	C410	C449	C448	C225	C306	C168	C447	C445	C443	C442	C440	C436	C435	C434	C484	C481
C518	C517	C516	C516	C328	C324	C502	C511	C117	C284	C302	C409	C405	C401	C402	C129	C223	C409	C24	C557	C317	C423	C466	C485	C481	C66	C154	C479	C78	C181	C560	C143	C472	C475	C166	C144
C546	C545	C544	C548	C542	C541	C37	C540	C338	C356	C335	C404	C353	C389	C388	C32	C332	C16	C348	C394	C331	C390	C399	C155	C100	C390	C337	C506	C217	C322	C311	C334	C322	C331	C330	C519
C581	C580	C576	C577	C574	C572	C571	C465	C570	C569	C557	C46	C7	C169	C56	C517	C104	C472	C53	C567	C566	C565	C519	C309	C153	C61	C561	C559	C552	C551	C510	C548	C541			
C1	C4	C12	C36	C68	C127	C129	C300	C5	C406	C302	C9	C598	C15	C315	C18	C5	C598	C292	C22	C388	C390	C389	C387	C28	C222	C388	C390	C170	C381	C384	C383	C382			

Figure 107: Benchmark 10 - Step 3/7 - Flip cells

C71	C406	C377	C270	C578	C701	C65	C68	C60	C53	C77	C54	C48	C47	C45	C48	C42	C46	C35	C34	C36	C33	C31	C19	C17	C11	C14	C10	C9	C6	C481	C97				
C123	C128	C122	C121	C130	C154	C124	C119	C118	C116	C113	C111	C110	C109	C108	C102	C98	C97	C96	C95	C93	C92	C91	C82	C81	C80	C77	C75	C74	C73	C72	C71	C64			
C185	C184	C183	C182	C178	C176	C175	C174	C173	C169	C164	C162	C160	C113	C151	C158	C156	C153	C149	C148	C145	C141	C138	C136	C135	C134	C133	C132	F15	C385	C446					
C221	C201	C470	C238	C404	C218	F14	C368	C548	C20	C477	C217	C215	C212	C209	C207	C203	C202	C201	C314	C483	C338	C389	C214	C197	C195	C191	C484	C390	C213	C573	C192	C189	C186		
C254	C251	C151	C150	C248	C488	C163	C245	C162	C244	C240	C413	C401	C53	C501	C444	C467	C415	C241	C303	C456	C400	C414	C414	C467	C231	C62	C463	C311	C232	C231	C229	C228	C224		
C293	C292	C388	C387	C381	C381	C405	C320	C304	C410	C279	C278	C291	F15	C274	C271	C270	C272	C288	C287	C286	C283	C282	C281	C280	C289	C288	C287	C286	C285	C284	C283				
C315	C313	C298	C291	C270	C269	C255	C254	C253	C252	C251	C250	C259	C258	C257	C256	C255	C254	C253	C252	C251	C250	C249	C248	C247	C246	C245	C244	F15	C384	C446					
C343	C342	C341	C340	C343	C344	C342	C413	C401	C403	C397	C444	C467	C421	C341	C341	C340	C346	C345	C344	C343	C342	C341	C340	C340	C341	C341	C340	C341	C342	C343	C344				
C381	C380	C381	C382	C381																															
C408	C407	C406	C402	C403	C401	C402	F14	C386	C387	C407	C217	C215	C212	C209	C207	C203	C202	C201	C314	C483	C338	C389	C214	C197	C195	C191	C484	C390	C213	C573	C192	C189	C186		
C430	C429	C433	C424	C423	C421	C419	C199	C139	C194	C196	C198	C159	C1	C34	C392	C213	C131	C268	C247	C336	C267	C300	C306	C416	C384	C383	C382	C381	C380	C381	C381	C381	C381	C381	
C473	C471	C469	C31	C147	C115	C130	C465	C463	F14	C461	C114	C32	C460	C401	C455	C161	C342	C433	C188	C410	C449	C448	C225	C306	C168	C447	C445	C443	C442	C440	C436	C435	C434	C484	C481
C518	C517	C516	C516	C328	C324	C502	C511	C117	C284	C302	C409	C405	C401	C402	C391	C390	C389	C388	C387	C386	C385	C384	C383	C382	C381										
C546	C545	C544	C543	C542	C541	C37	C540	C338	C355	C348	C348	C348	C347																						
C581	C580	C576	C577	C574	C572	C571	C465	C570	C569	C557	C46	C7	C348	C465	C467	C461	C129	C223	C460	C24	C337	F14	C386	C213	C131	C271	C171	C467	C394	C393	C340	C447	C446		
C613	C611	C609	C608	C607	C606	C605	C604	C603	C602	C601	C600																								
C651	C650	C649	C648	C647	C646	C645	C644	C643	C642	C641	C640																								
C689	C688	C687	C686	C685	C684	C683	C682	C681	C680																										
C71	C70	C69	C68	C67	C66	C65	C64	C63	C62	C61	C60																								

Figure 108: Benchmark 10 - Step 4/7 - Add feed-throughs

C4	C408	C127	C377	C396	C278	C25	C38	C175	C244	C388	C360	C407	C429	C301	C338	C566	C382	C386	C174	C538	C322	C415	C481	C381	C338	C279	C337	C134	C45	C309	C178	C496	C412	C30	C397			
C5	C554	C68	C124	C398	C125	C38	C102	C171	C245	C110	C362	C408	C103	C301	C340	C367	C283	C387	C175	C551	C324	C478	C484	C382	C338	C280	C301	C131	C363	C178	C496	C412	C30	C397				
C59	C513	C388	C157	C381	C395	C402	C393	C351	F135	C40	C129	C192	C361	C338	C174	C413	C464	C380	C443	C397	C154	C394	C129	C158	C83	C10	C157	C476	C545	C533	C571	C385	C338	C192	C446			
C53	C301	C42	C470	C445	C404	C399	C101	C356	F154	C295	C38	C231	C546	C160	C30	C391	C471	C14	C38	C134	C111	C481	C19	C238	C303	C369	C89	C331	C144	C375	C494	C434	C296	C465	C213	C318	C373	
C479	C438	C485	C163	C385	C313	C410	C413	C451	C212	C83	C138	C307	C71	C444	C487	C405	C303	C456	C600	C36	C454	C108	C414	C178	C467	C348	C482	C311	F131	C363	C493	C409	C455	C311				
C546	C401	C519	C130	C369	C304	C428	C400	C377	C391	C310	F131	F129	C512	C195	C388	C122	C1	C92	C54	C37	C160	C403	C133	C319	C396	C211	C357	C461	C400	C489	C447	C399	C87	C233	C334	C421		
C411	C199	C364	C361	C113	C398	C250	C370	C157	C309	C109	C109	C173	C34	C171	F12	C403	C14	C150	C383	C103	F132	C54	C384	C144	C194	C72	C105	C138	C491	C215	C18	C32	F131	C391	C365	C418		
C251	C171	C183	C325	C110	C339	C35	C432	F131	C366	C140	C333	F131	C173	F131	C289	C146	C387	C379	C73	C152	F132	C366	C323	F132	C20	F131	C70	C343	F132	C452	C344	C180	C75	C427	C141	C489	C217	F131
C381	C255	C235	C36	C164	C372	C93	C54	C17	C437	F131	C300	C322	C181	F134	F131	C300	C324	C137	C55	F131	C199	C493	C466	C303	C16	F132	C316											
C552	F130	C32	C397	C401	F131	C489	C420	C391	C364	C348	C381	C383	C360	C118	C159	C145	C41	C111	C76	C189	C184	C189	C201	C303	C139	C303	C341	C413	C112	C457	C174	C304	C469	C341	C541	C67		
C334	C198	C275	C419	C381	C74	C124	C133	C136	C158	C164	C234	C263	C227	C298	F131	C13	C366	C404	C401	C324	C132	C355	C397	C171	C377	C19	C246	C346	C95	C340	C469	C44	C337	C355	C303	C348		
C367	C409	C133	C190	C254	C339	C181	C194	C113	F134	C40	C196	C159	C1	C384	C392	C223	C131	C388	C247	C82	C236	C367	C300	F133	C356	C356	C77	C179	C148	C151	C228	C387	C142	C349	C391			
C584	C31	C568	C147	C542	C115	C398	C150	C400	F131	C416	C114	C368	C52	C30	C149	C221	C388	C136	C256	C156	C116	C43	C151	C161	C393	C342	C446	C183	C451	C12	C306	C225	C358	C306	C376	C168		
C541	C151	C533	C117	C471	C394	C429	C129	C371	C223	C337	C24	C276	C537	C381	F131	C308	C21	C96	C86	C317	C173	C66	C246	C154	C305	C78	C138	C107	C421	C500	C448	C143	C495	C166	C353	C144		
C572	C37	C548	C426	C321	C35	C473	F131	C426	C38	C379	C380	C388	C32	C278	C33	C97	C174	C342	C385	C133	C29	C16	C307	C348	C139	C34	C403	C90	C449	C99	C155	C356	C100	C365	C27			
C593	C40	C579	C165	C527	C357	C461	C46	C441	C7	C394	C56	C303	C115	C45	C118	C189	C360	C391	C24	C113	C91	C6	C327	C104	C369	C412	C417	C33	C401	C59	C110	C209	C411	C531	C369	C61		
C570	C300	C544	C8	C316	C302	C463	C9	C411	C15	C391	C315	C388	C31	C307	C3	C382	C316	C156	C238	C191	C119	C47	C18	C293	C5	C333	C22	C398	C388	C442	C328	C485	C202	C322	C338	C39	C561	C170

Figure 109: Benchmark 10 - Step 5/7 - Even out the row length

C4	C137	C406	C377	C396	C398	C66	C134	C307	C279	C138	C381	C401	C405	C303	C509	C374	C566	C388	C392	C400	C407	C360	C388	C344	C175	C98	C35	C270	C178	C496	C303	C412	C397					
C5	C368	C554	C124	C398	C390	C70	C135	C388	C280	C189	C382	C484	C478	C354	C501	C375	C587	C383	C567	C440	C303	C408	C380	C310	C345	C176	C301	C125	C347	C341	C347	C346						
C59	C388	C513	C157	C381	C395	C502	C517	C120	C192	C83	C48	C354	C400	C397	F135	C474	C393	C448	C138	C229	C433	C374	C138	C361	C130	C31	C394	C464	C545	C571	C353	C352	C359	C446				
C381	C316	C301	C470	C124	C441	C101	C482	F134	C148	C395	C160	C395	C113	C332	C10	C477	C14	C38	C111	C314	C481	C389	C238	C303	C193	C311	C375	C424	C396	C318	C373							
C479	C414	C488	C163	C385	C381	C383	C381	C63	C451	C411	C410	C312	C316	C307	C444	C474	C405	C303	C456	C600	C487	C454	C108	C178	C71	C132	C368	C311	C345	C341	C347	F131						
C57	C346	C403	C130	C319	C428	C343	C40	C377	F131	C269	C391	C332	F131	C12	C11	C382	C383	C122	C154	C334	C322	C314	C308	C160	C469	C400	C357	C447	C401	C211	C489	C399	C487	C334	C333	C421		
C411	C364	C198	C361	C113	C398	C250	C182	C370	C189	C209	C109	C273	C34	C171	F134	C403	C138	C21	C96	C86	C317	C173	C66	C246	C154	C305	C78	C138	C107	C421	C500	C448	C143	C495	C166	C353	C144	
C551	C183	C172	C325	C366	C110	C329	C432	F131	C40	C311	F131	C187	C373	C179	C259	C146	C387	C73	C152	C355	C325	C388	C130	C343	C135	C248	C315	C462	C342	C347	C345	C346	F131					
C501	C93	C36	C17	C164	C354	C355	C437	F131	C177	C300	C322	C181	F134	C57	C36	C389	F131	C313	C37	C55	F131	C199	C493	C466	C303	C16	F132	C316										
C552	C481	F130	C325	C397	C391	C486	F131	C384	C430	C381	C411	C111	C139	C76	C365	C341	C381	C288	C40	C118	C318	C322	C389	C141	C345	C329	C319	C355	C331	C341	C347	C346	C342	C341				
C334	C74	C198	C275	C419	C381	C735	C133	C136	C158	C164	C234	C263	C227	C298	F131	C13	C366	C404	C401	C324	C132	C355	C397	C19	C246	C346	C95	C340	C469	C44	C337	C355	C303	C348				
C367	C231	C49	C190	C254	C339	C181	C194	C113	F134	C40	C196	C159	C1	C384	C392	C223	C131	C388	C247	C82	C236	C367	C300	F133	C356	C356	C77	C179	C148	C151	C228	C387	C142	C349	C391			
C584	C31	C68	C147	C542	C398	C115	C150	C460	F131	C416	C211	C40	C52	C32	C114	C368	C388	C363	C366	C356	C306	C385	C313	C481	C161	C342	C235	C123	C318	C318	C318	C318	C318	C318				
C547	C320	C311	C117	C429	C394	C378	C129	C223	C319	C20	C310	C246	F131	C537	C317	C276	C86	C446	C201	C154	C107	C78	C124	C66	C148	C560	C420	C11	C485	C355	C166	C144						
C572	C331	C548	C473	C37	C35	C468	F131	C426	C389	C31	C383	C278	C179	C16	C318	C97	C174	C342	C303	C133	C29	C348	C94	C446	C403	C90	C99	C217	F131	C336	C336	C336	C336	C336				
C593	C40	C145	C327	C466	C357	C46	C441	C112	C189	C308	C384	C315	C313	C7	C16	C104	C392	C34	C13	C313	C91	C6	C327	C45	C369	C417	C472	C313	C309	C209	C345	C569	C353	C41				
C570	C344	C300	C8	C316	C303	C302	C9	C411	C371	C15	C315	C308	C31	C307	C3	C381	C316	C156	C238	C191	C47	C18	C313	C333	C395	C22	C322	C338	C368	C38	C202	C322	C338	C361	C39	C170		

Figure 110: Benchmark 10 - Step 6/7 - Pull cells in the rows closer together

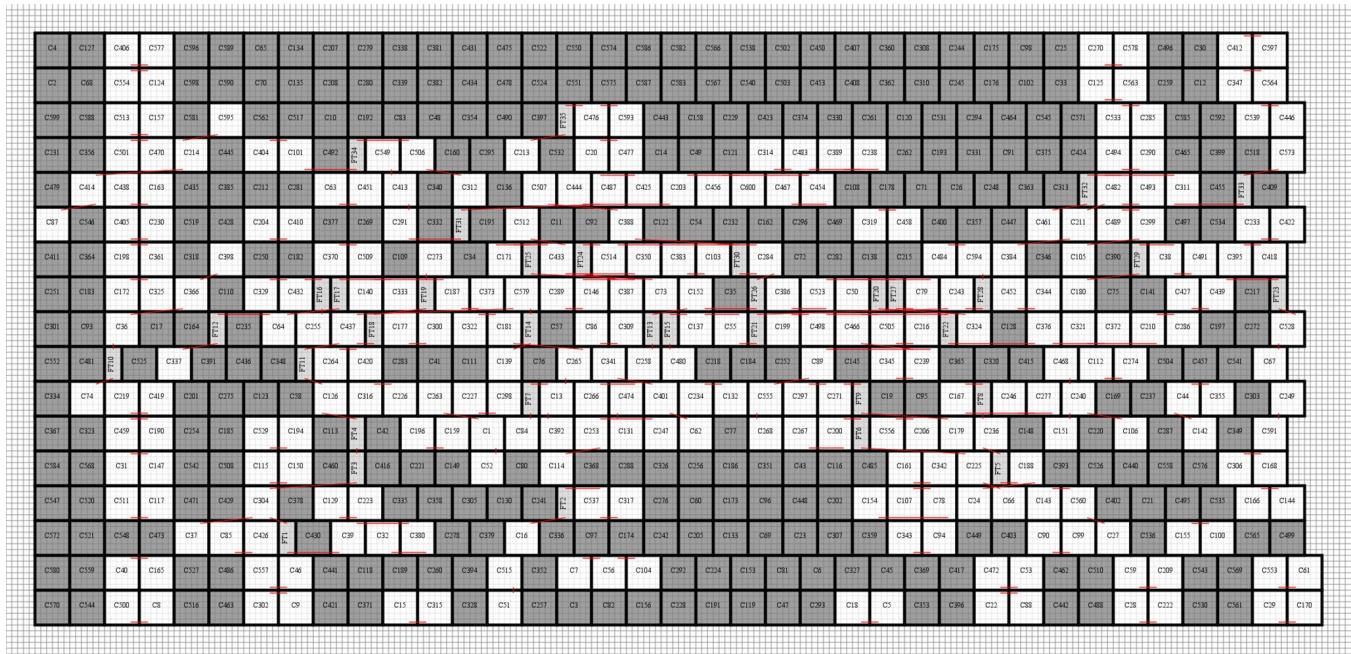


Figure 111: Benchmark 10 - Step 7/7 - Move feed-throughs to optimal location

4.38 Benchmark 10 - Final Routing

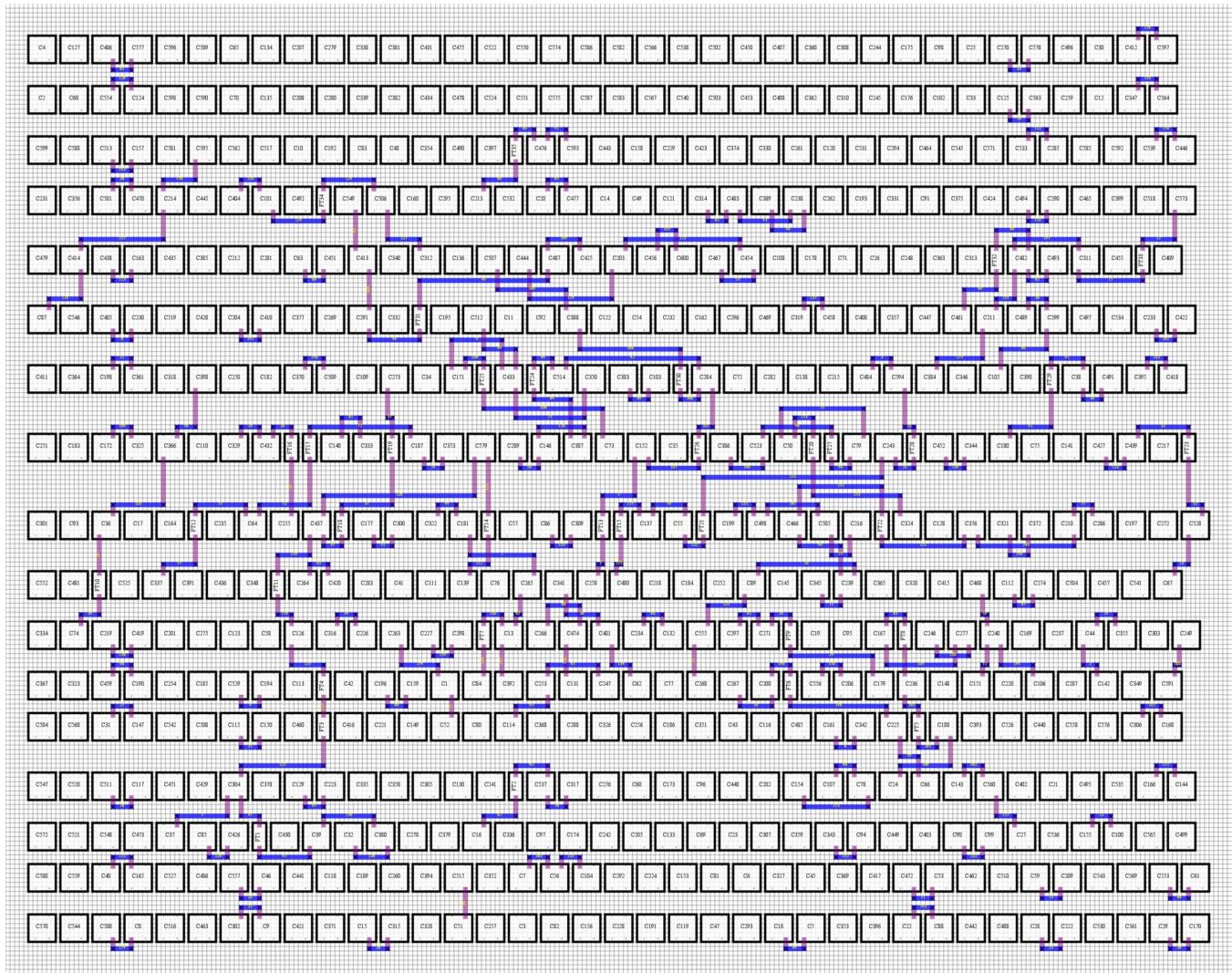


Figure 112: Benchmark 10 - Final routing

4.39 Benchmark 10 - Magic Screenshot

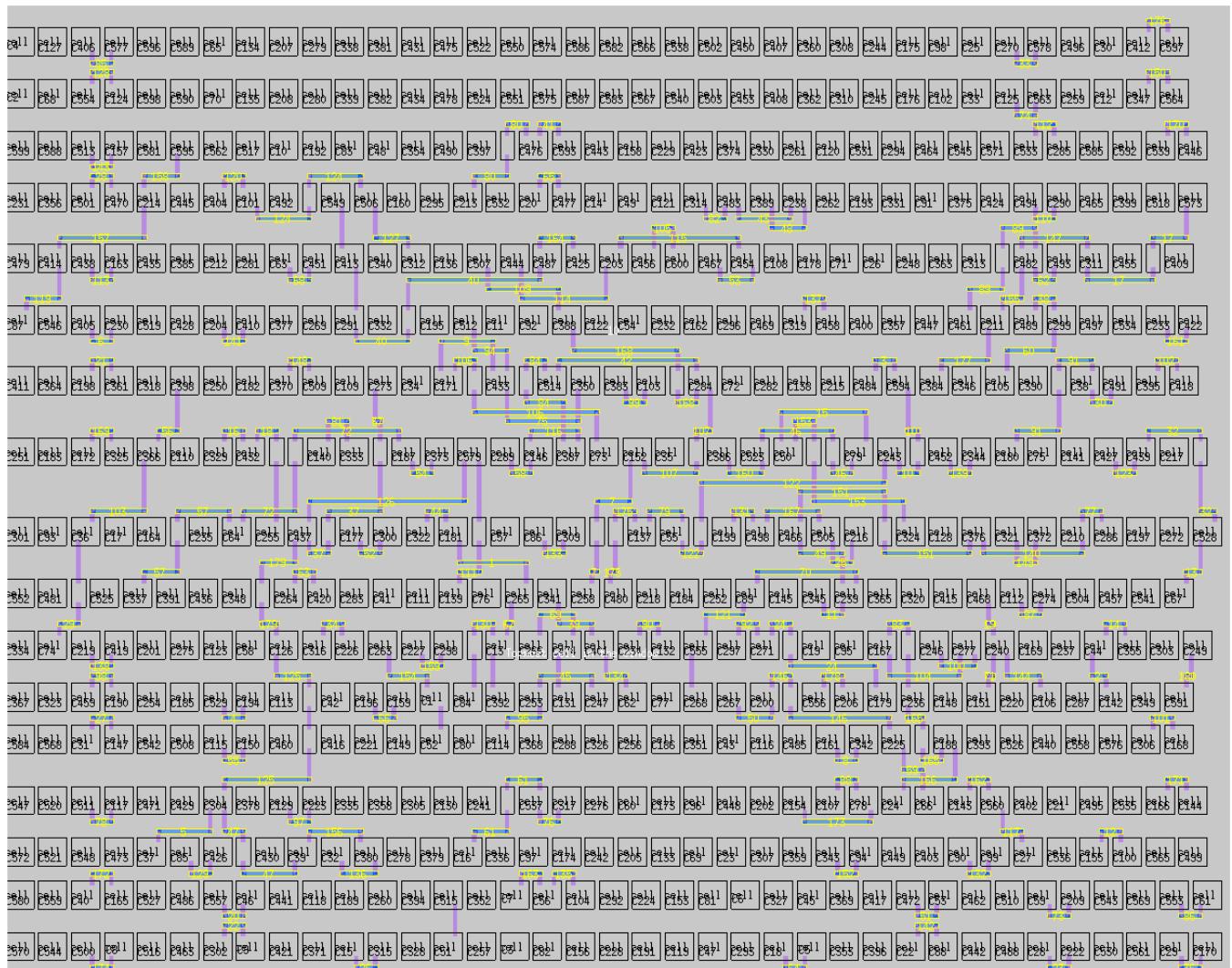


Figure 113: Benchmark 10 - Magic Screenshot

4.40 Benchmark 10 - Log file

```
1 Number of cell: 600
2 Number of nets: 180
3
4 Placement
5 _____
6 Initial grid size: 36 17
7 Number of feed through cells: 35
8
9 Routing
10 _____
11 Number of vias: 404
12 Total number of tracks: 42
13 Total wirelength: 2881
14
15 Size
16 _____
17 Total width: 258
18 Total height: 202
19 Total area: 52116
20 Squareness (width/height): 1.277228
21
22 Time
23 _____
24 Place time: 0.138171s
25 Route time: 0.004349s
26 Total time: 0.159355s
27
28 Memory used: 544.000000 kB
```

Listing 10: Benchmark 10- Log

5 Retrospective

Overall, we are pretty satisfied with both the execution speed and quality of our results. All of our execution times with the exception of benchmark 7 are under half a second. We also feel that our memory usage is minimal as we only store one vector of actual cell objects and all other data structures just contain pointers back to the original objects. We also make heavy use of pass by reference to avoid needlessly copying big data structures into functions.

In terms of placement and routing quality there is always room for improvement. It's easy to visually look at almost any result and see ways that it could be improved but how to translate these improvements into algorithms is not always so obvious. We believe, however, that we were able to handle a lot of specific cases (like moving the feed-throughs to the other side of the cell) which in the end resulted in superior layouts. For the force directed placement algorithm it is hard to tell if the results you get are truly the best, especially when the circuit is so large you cannot visually comprehend it. We feel like we did our best analyzing it empirically and tuned it to best suit our benchmark files. One of the main problems with the force directed algorithm is that it tends to pull all the cells into the center of the layout. We tried to fix this by redistributing the unconnected cells evenly through the layout and then re-force directing the rows but it is still not always ideal. One interesting method to pursue would be using a combination of min-cut / quadratic placement and then force directing the sub divisions (or vice versa). If we were to min-cut horizontally we might be able to reduce some nets that currently span multiple rows.

6 Appendix

Benchmark	Num Cells	Num Nets	Grid Width	Grid Height	Squareness
1	45	45	7	7	0.72449
2	90	90	10	9	0.736486
3	180	180	14	13	0.676724
4	360	360	19	19	0.702857
5	720	720	27	27	0.612457
6	1000	1000	32	32	0.605988
7	500	1000	23	22	0.645203
8	900	720	30	30	0.630648
9	800	360	29	28	0.634349
10	600	180	25	24	0.67148
1	45	45	8	6	0.820225
2	90	90	27	8	0.984496
3	180	180	12	12	0.755981
4	360	360	15	18	0.740413
5	720	720	20	26	0.595577
6	1000	1000	28	31	0.561298
7	500	1000	33	21	0.614935
8	900	720	24	29	0.637652
9	800	360	32	27	0.685294
10	600	180	30	23	0.75
1	45	45	9	5	1.078947
2	90	90	13	7	1.20339
3	180	180	17	11	0.921875
4	360	360	22	17	0.833333
5	720	720	29	25	0.681979
6	1000	1000	34	30	0.583232
7	500	1000	25	20	0.646465
8	900	720	33	28	0.672032
9	800	360	31	26	0.731629
10	600	180	28	22	0.773946
1	45	45	12	4	1.779661
2	90	90	15	6	1.623762
3	180	180	18	10	1.091429
4	360	360	23	16	0.84127
5	720	720	30	24	0.742911
6	1000	1000	35	29	0.624372
7	500	1000	27	19	0.692828
8	900	720	34	27	0.711864
9	800	360	32	25	0.790323
10	600	180	29	21	0.869748
1	45	45	15	3	2.923077
2	90	90	18	5	1.864583
3	180	180	20	9	1.322368
4	360	360	24	15	0.920382
5	720	720	32	23	0.75098
6	1000	1000	36	28	0.699068
7	500	1000	28	18	0.715323
8	900	720	35	26	0.694501
9	800	360	34	24	0.810289
10	600	180	30	20	0.991111
1	45	45	23	2	4.945946
2	90	90	23	4	2.626667
3	180	180	23	8	1.391608
4	360	360	26	14	0.986532
5	720	720	33	22	0.848671
6	1000	1000	38	27	0.674667
7	500	1000	30	17	0.763196
8	900	720	36	25	0.832599
9	800	360	35	23	0.883333
10	600	180	32	19	1.074074

Table 3: Squareness Trial Data