(10) Update on Ghost Busters & Getting no unconnects "as SHORT as POSSIBLE"

Hugh Allen

CopperCAD Design
www.coppercad.com

September 2004

International
Cadence
Usergroup

Ghost Content

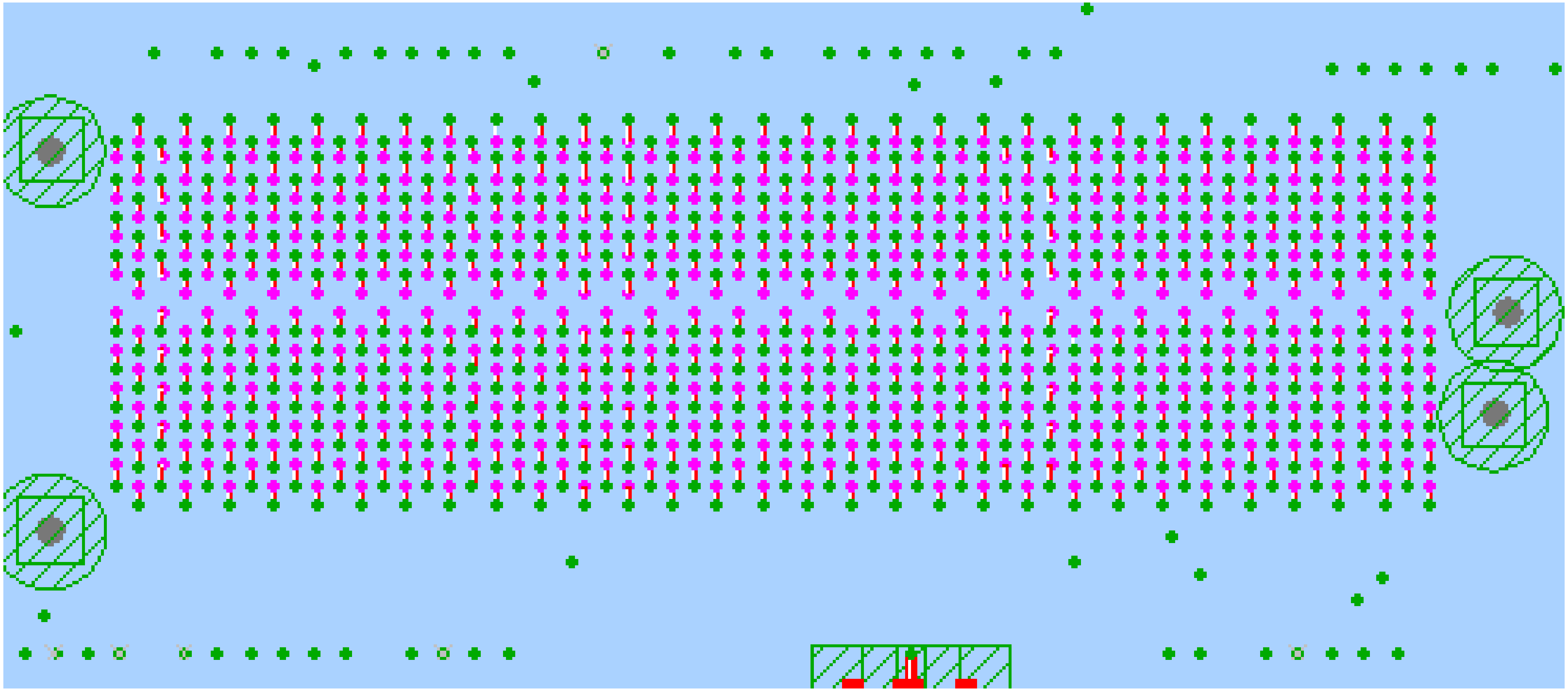
- A look at some current Specctra **"things that go thump in the night**“ **And some “Known exorcisms”**
- The connector that auto connects fine in 14.1 not in 14.2 for **no earthly reason Fixed in rel. 15**
- **Snowballing Poly shapes.**
- Strange Bends and flow disruption surprise
- Slow (simple solution) (secret solution)
- Specctra cannot fully utilize all possible large bga exits (45 degree routing needed)
- Bus command divorcing pairs.
- Miter not mitering to meet max allowed, when min rule exits as well as max.

Getting no unconnects and “as short as possible” etc.

- Making Specctra connect the unconnects.
- Identifying trapped unconnects early.
- Cleaning the best result rather than the last one from a route series in spite of snowballing poly wires (planes).
- Making Specctra reduce the last “few crossing conflicts” in a stall situation automatically.
- Making Specctra shorten up sloppy routing including hairpin loops. (“wringer”)
- (a set of do file sequences)

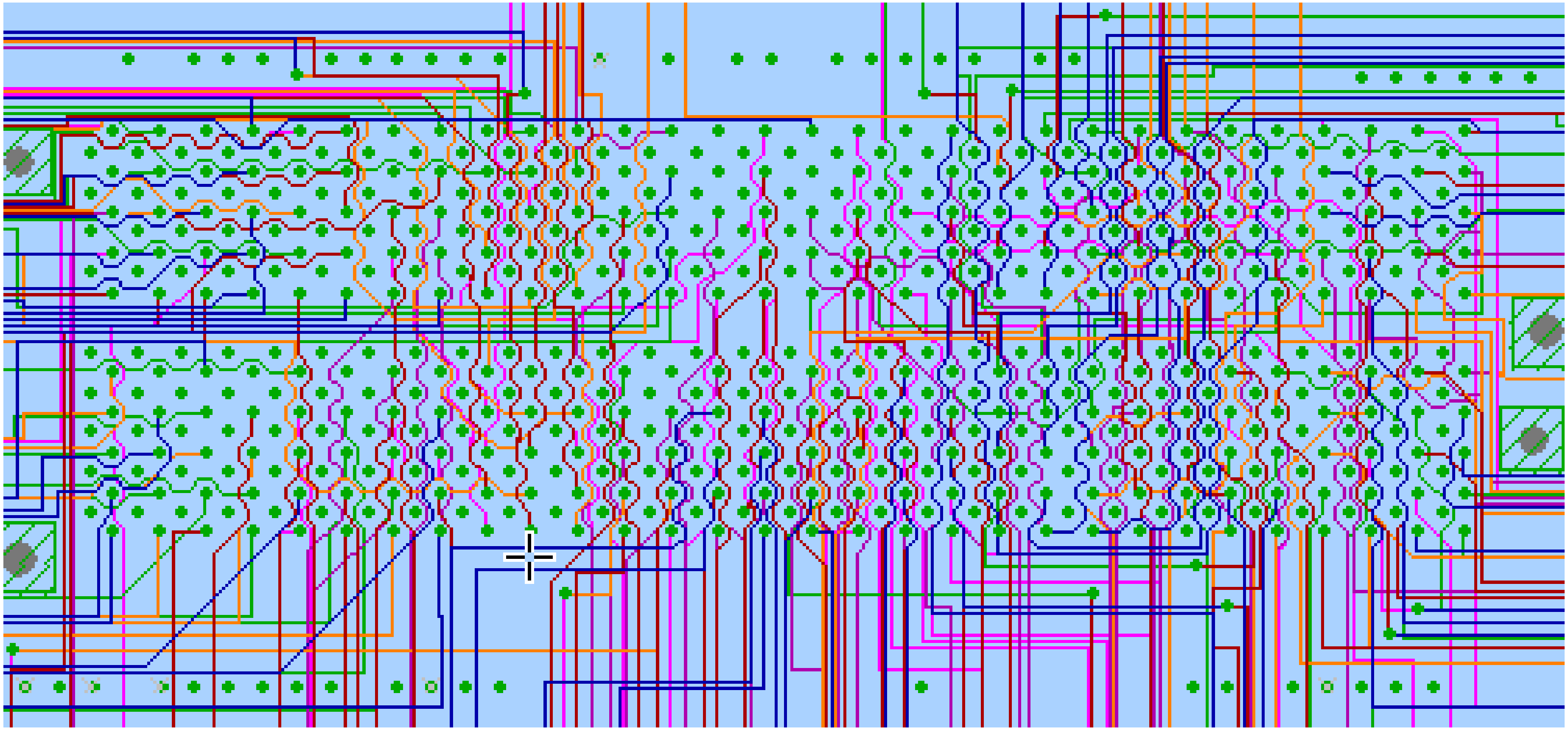
CONNECTOR OK IN 14.1 NOT IN 14.2 YES 15.O

- **NEXLEV CONNECTOR 600 pin mother to daughter**



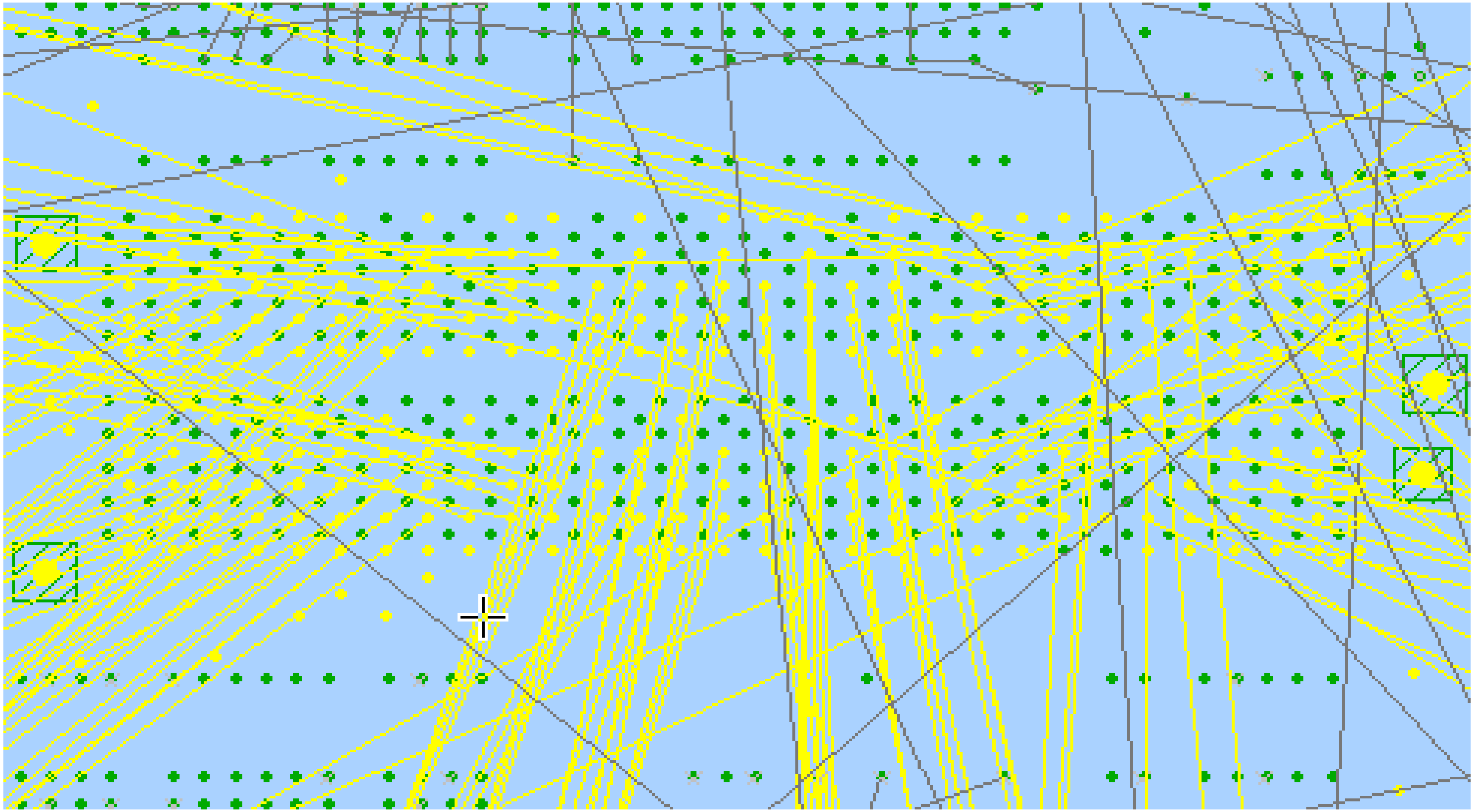
CONNECTOR OK IN 14.1 NOT IN 14.2 →15.O OK

- NEXLEV OK IN 14.1 AND 15.0 100% 2 minutes



CONNECTOR OK IN 14.1 NOT IN 14.2 →15.O OK

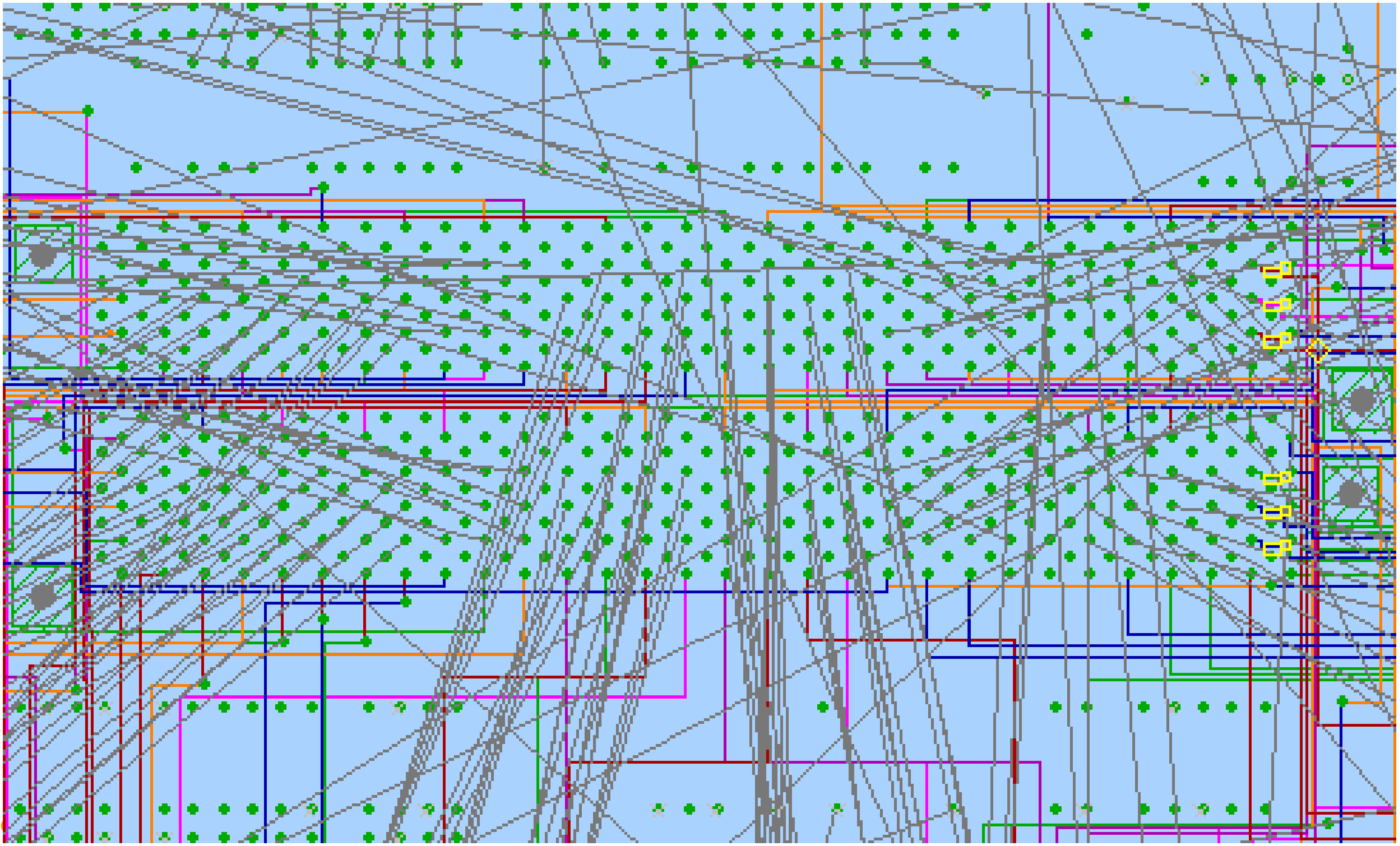
- NEXLEV in 14.2 auto fails will route manual



CONNECTOR OK IN 14.1 NOT IN 14.2 →15 OK

- **NEXLEV in 14.2 auto fails**

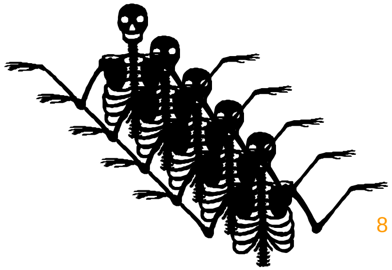
FIXED IN 15



SNOWBALLING POLY SHAPES

- **Watch out for Specctra duplicating positive poly shapes if a wire/route/ses file is written then read in.**

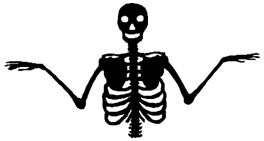
– Steps in do file	instances of each shape
– read dsn	1
– Write, delete all wires, read	2
– Write, delete all wires, read	4
– Write, delete all wires, read	8
– Write, delete all wires, read	16 etc. etc.



SNOWBALLING POLY SHAPES

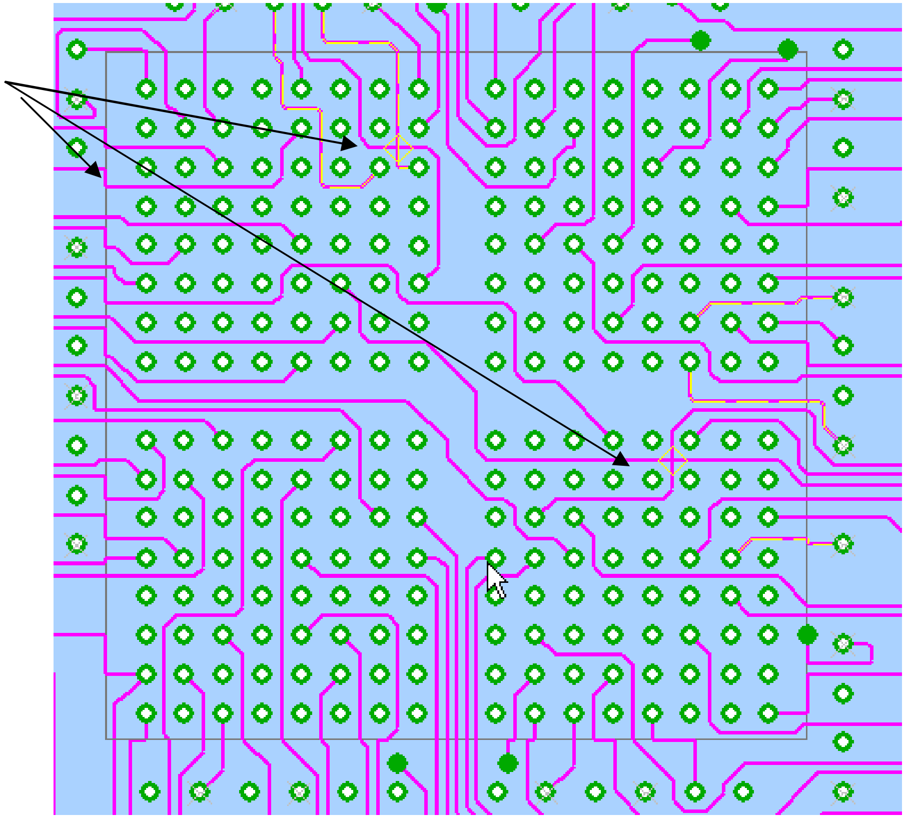
- Watch out for Specctra duplicating positive poly shapes if a wire/route/ses file is written then read in.
- These duplicates dramatically slowdown the checking processing in Specctra, and in Allegro when loaded back in.

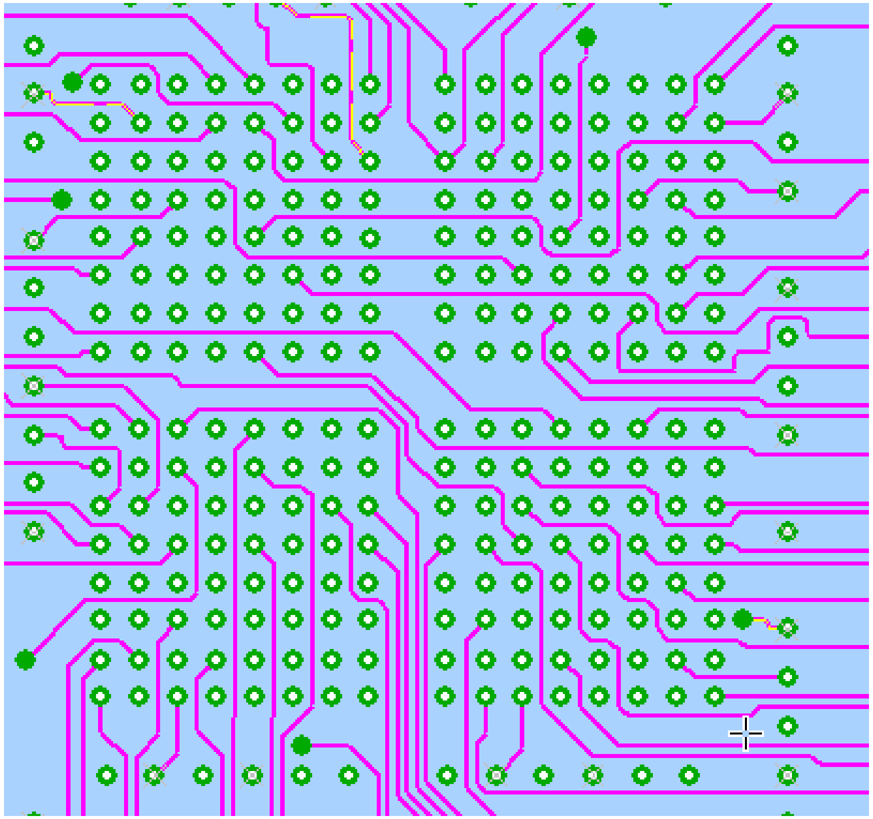
  – Workaround = write, then unprotect poly, then delete poly, then read. Works with all wire files including read wire from bestsave.



Strange Bends and flow disruption surprise

- Accidental regions degrade routing, miter, and trombones!





Strange Bends and flow disruption surprise

- Regions can be useful but must be carefully managed
  - All rule areas defined in Allegro for whatever purpose migrate into Specctra as regions often on all layers. (15 is somewhat better)
  - If no special clearance rules are defined they inherit the lower priority rules and disrupt things for no good reason.
  - The boundary of a region creates a new segment for each trace that crosses it – one for the rules inside and another for the rules outside.
  - Routing flow, miter processing and tromboning are disrupted.

Region control

International
Cadence
Usergroup

- Make sure all regions that do NOT have a specific routing purpose in Specctra are deleted.

- Make sure that required regions are ONLY present on the layers that need them. Example a bga requires special smd to via or through hole via to via clearance
  - ONLY a region on the top layer is required for smd clearance.
  - And then only if the fanouts are being done in Specctra.
  - If done in Allegro fanouts are protected in Specctra so the region is not needed unless we expect specctra to utilize vacant via sites in the via array. (a useful case)

Region control

International
Cadence
Usergroup

- If we have a set of diff pairs that require a clearance reduction in a bga via array.
  - If the pairs are constrained to only certain layers then the regions are only required on those layers and in some cases only in some quadrants of the bga via array.

- A similar technique can be applied to multi row connectors with the bonus of providing guidance by limiting the region definition to avoid wandering around unnecessarily.

- Watch out for Specctra using regions to cheat on wide trace power. Job for region class rule.

Region control

- On very congested designs it sometimes is necessary to define temporary clearance rules that will allow access to the bga via array for the hook up stage in the auto routing.

- Later once Specctra has found the hook up bga channels with crosses/clearances down, regions rules are defined and clearance rules redefined and routing continues.
- **Change_width_by_rule** option no_violation or force option (changes them all)

- Careful using a temp width in this way because Specctra will not necessarily change width with a rule change and has no check for width violations. Force option with Visual check is recommended.

Region Improvements

- The level of disruption seems to have improved a bit over the last few revisions.
- Spif is less prone to passing these than it used to be but they still can cause trouble.

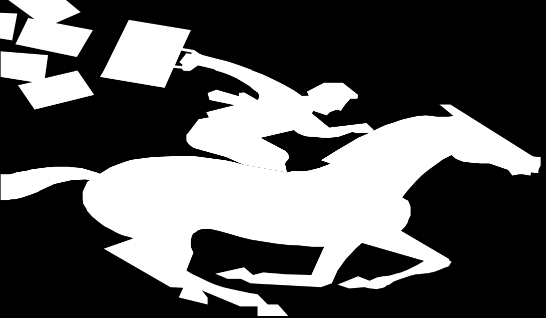
- The addition of region net rules, region class rules and region class_class rules provides solutions to some tricky problems like the power one, and high voltage special clearances that regions on their own messed up on.

Region has a new use in version 15

- We will have Specctra popping vias through small planes on the signal layers & plowing into these planes with traces!
- How to keep the Swiss cheese from becoming all hole and no cheese? And cut up in unconnected strips?
- Defining a region with brutal clearance via to via, wire max stagger, region class_class wire to wire will help avoid “all hole and no cheese.”
- **We really do need an additional set of clearance rules for for poly wires by net (WYSIWYG planes)!**

SLOW

- VERY DENSE HEAVILY CONSTRAINED DESIGN:
  - 4” X 8” Area/EIC= 0.049 sq. in
  - Components=2052 SMDs=1915 Pin Count: 10483
  - Start vias **15106 (micro and buried)**
  - MICRO VIAS IN PAD OR TANGENTIAL FOR almost ALL
  - During early feasibility experimental placement stage of routing Specctra running just fine on **1.8 Gig** machine
  - 10 sec.for dsn load 1 sec. check (type routing) and 5 min for early routing passes dropping reasonably. (ETA 25 -30 hrs)
  - Then a some updates & the dsn load dragged out, check (type routing) 2 minutes then 5 min. then 8 minutes with the first routing to 21 minutes (fanout pass taking 32 min)! (ETA 150 hrs)

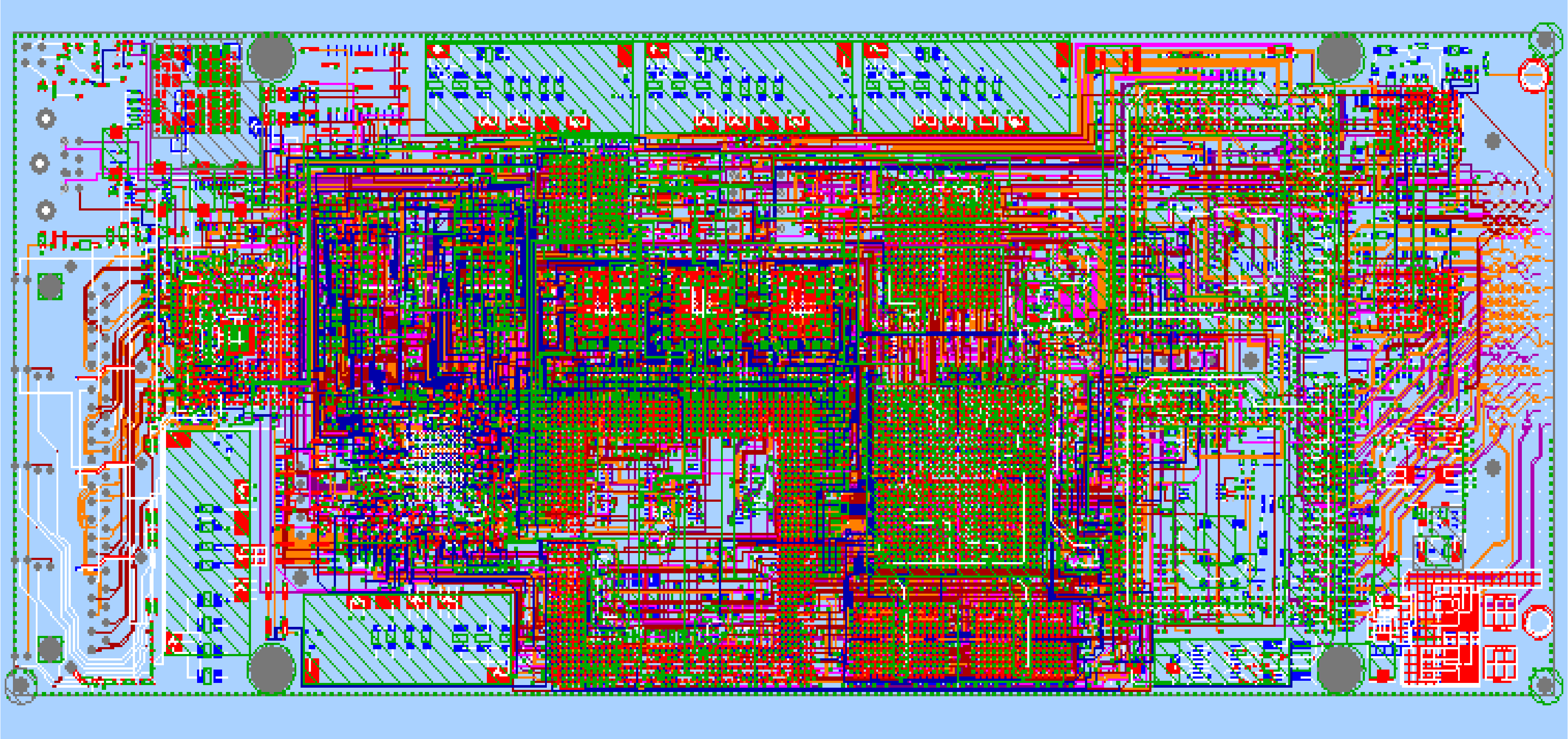


-simple

- VERY DENSE HEAVILY CONSTRAINED DESIGN:
  - In startup menu -> more options , click **simplify polygons**
  - Or on command line add option **–simple**
- On this design speed up by **8 to 10 times**
- **Able to load the design on 0.5 gig unix machines in finite time. (not infinite time)**
- Only works on designs with some shape problems.
- Other undocumented Allegro commands that help!
- **set spif_no_plane_windows**
- **set spif_keepout_for_shapes**
- **set SPIF_USE_WIRE_KEEPOUT**

SIMPLE
PIECE OF CAKE

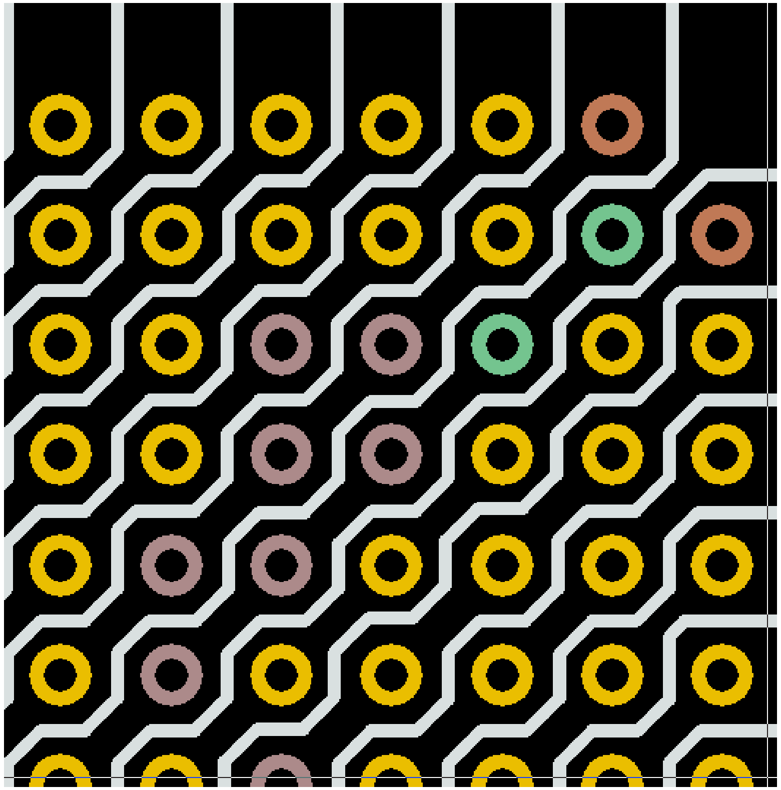
- 18 layer Area/EIC= 0.049 sq.in



large bga exits (45 degree routing needed)

- Specctra cannot fully utilize all possible large bga exits (45 degree routing needed)
- This NEED is detailed in my other talk.

Imagine 2 between!

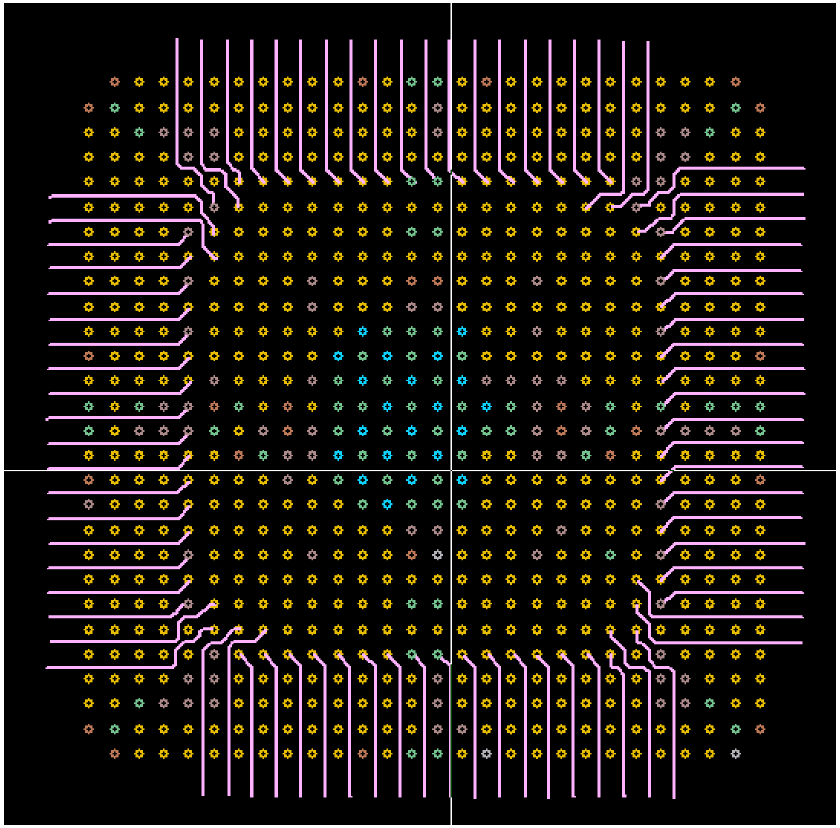


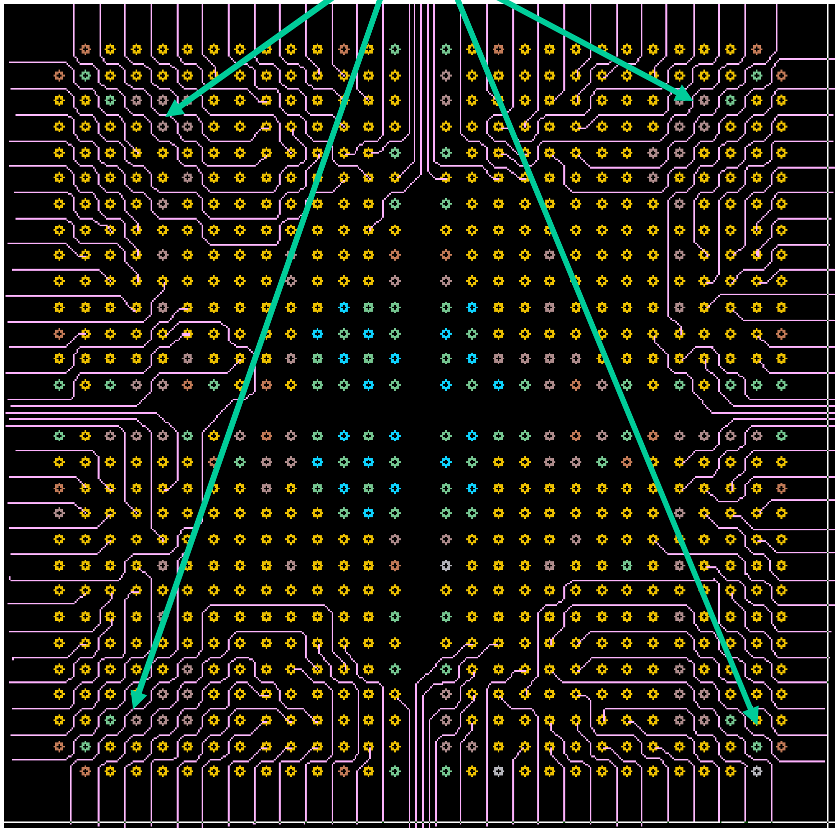
Miter plus turbo whatever simply does NOT do it!

large bga exits (45 degree routing needed)

International
Cadence
Usergroup

- This NEED is detailed in my other BGA talk.
- Utilization of all exits on ALL layers requires 45 degree





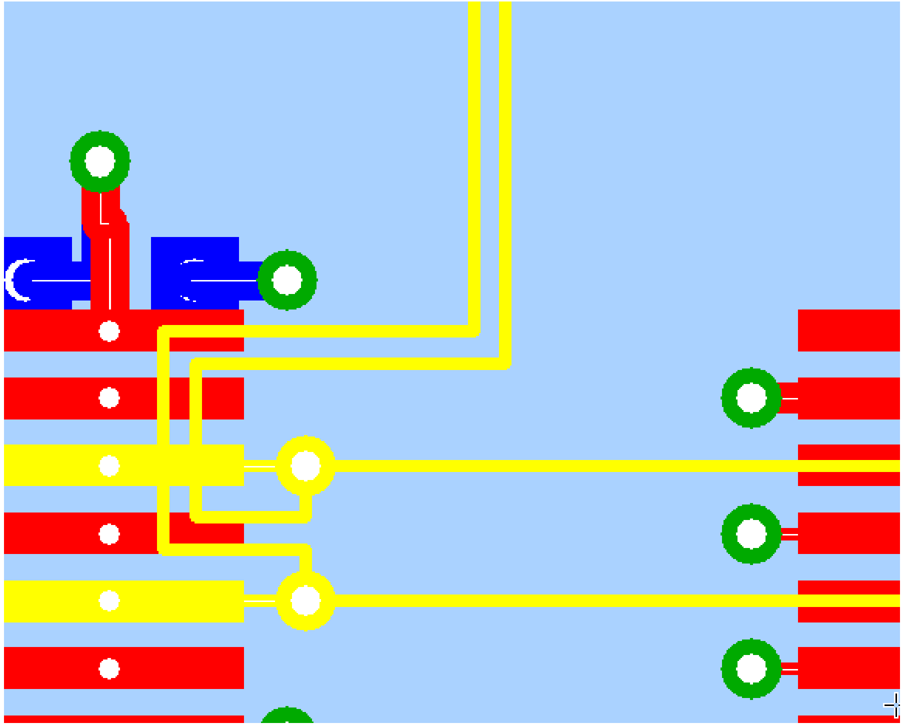
large bga exits (45 degree routing needed)

- NO REALLY GOOD SOLUTION IN THE CONTEXT TO AUTO ROUTING.
  - A lot of manual intervention and cleanup which are both very time consuming.
- Run in manual exits & protect for the first stage of routing, if unprotected these tend to act about as badly as bga fanout dogbones when they are unprotected.
- **Impact**:
  - **Increased layers (cost or feasibility)**
  - **Increased design time (cost or feasibility)**
  - **Delayed application of new technology (2000 to 3000 pin BGA’S)**

Bus command divorcing pairs.

- Ever wonder why specctra routed some pair members without regard for gap and **even on different layers**?
  - You used the Bus command to get things stared, that is the reason.

If you found it, you did so visually, or by building a board and having it fail functionally. There is no drc check in Specctra or (in Allegro?).



Bus command divorcing pairs.

–Bus (fast connect in-line pins/vias)

–WARNING DO NOT USE ON DESIGNS CONTAINING DIFF PAIRS, OR THAT MIGHT IN THE FUTUR HAVE DIFF PAIRS DEFINED.

Miter not mitering to meet max allowed

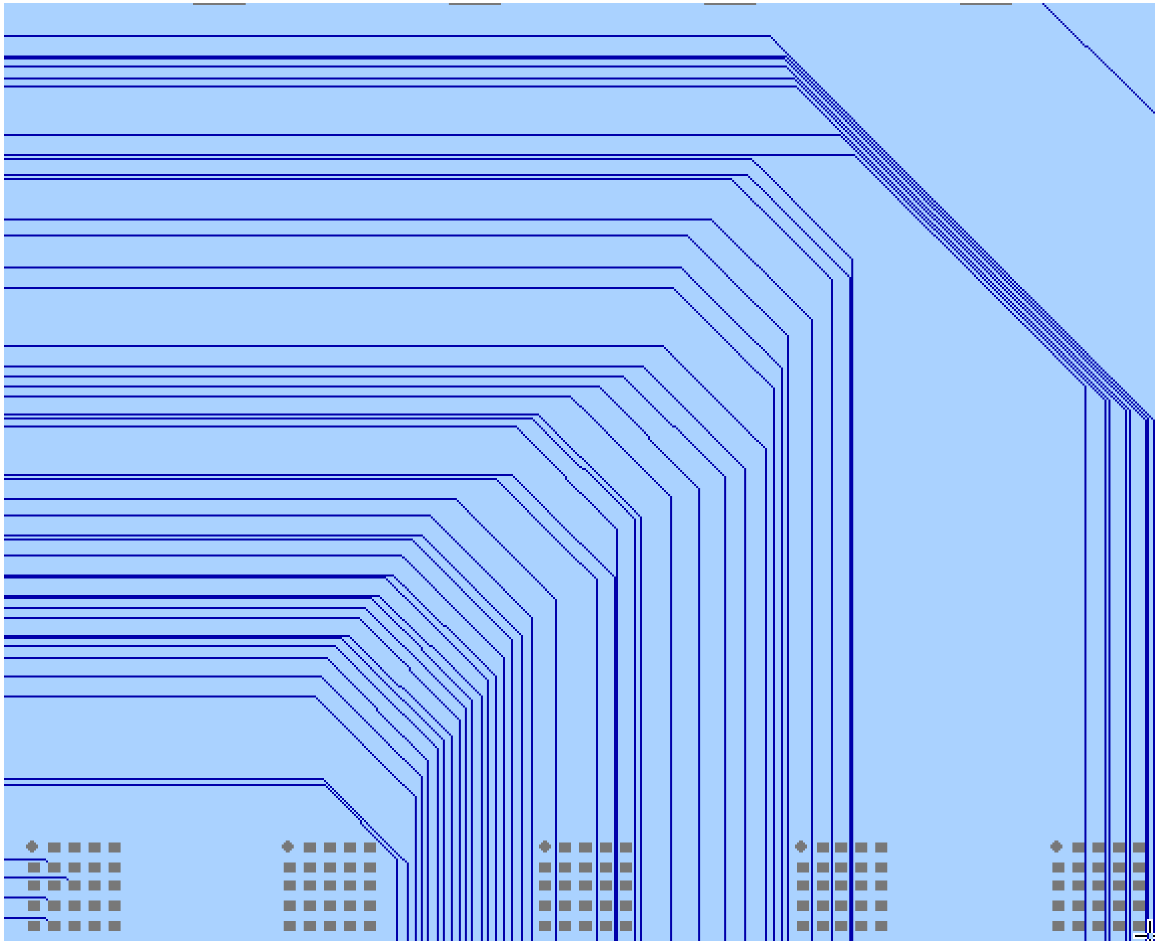
- **Miter not mitering to meet max allowed, <u>when easy min. rule exits</u> as well as max.**

  - **This can be isolated by running 3 tests on the same design and do file except:**

1. **No length rules at all**
2. **Max. allowed length limited**
   circuit selected (length 1.05 (type ratio))
3. **Max. allowed length limited plus min. required circuit selected (length 1.05   0.7 (type ratio))**

Miter not mitering to meet max allowed

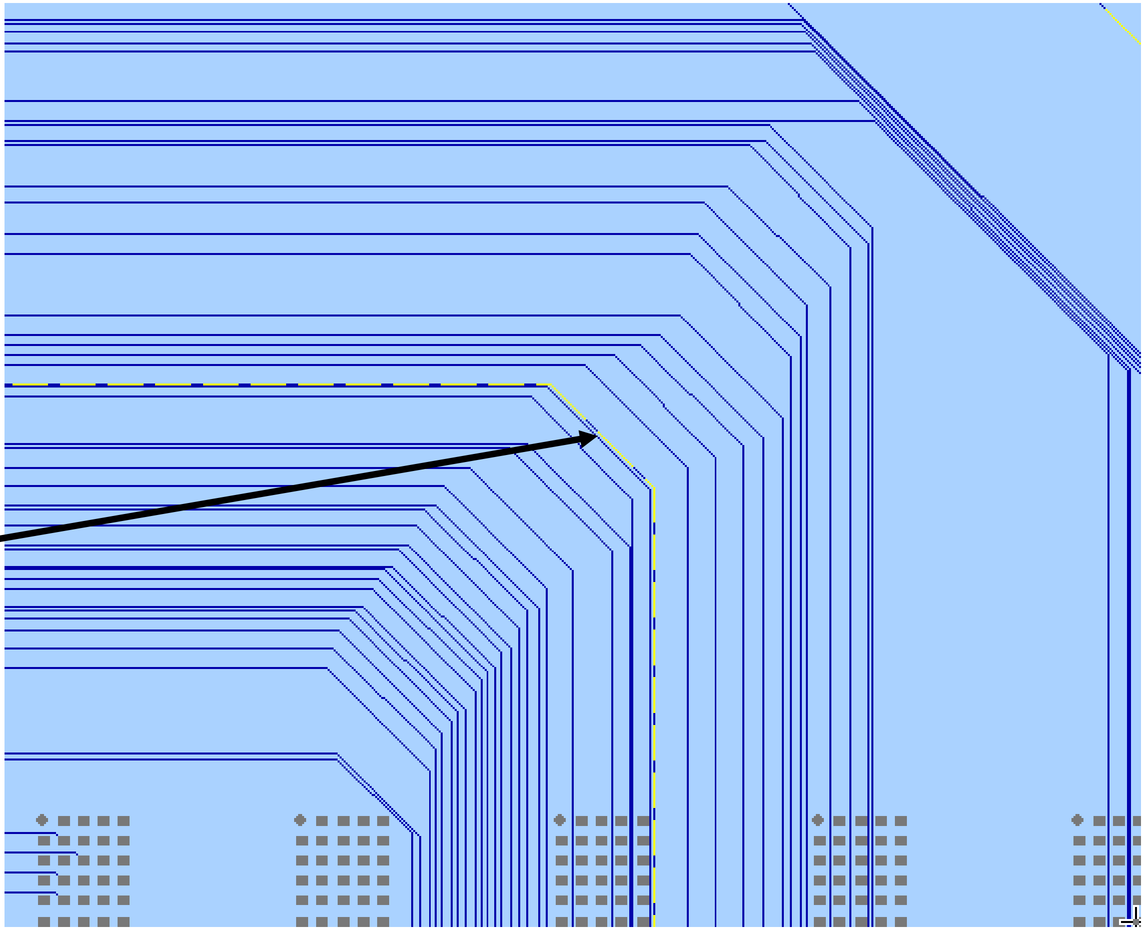
1. No length rules at all



Note the miters are pretty good

Miter not mitering to meet max allowed

2. circuit selected (length 1.05 (type ratio))



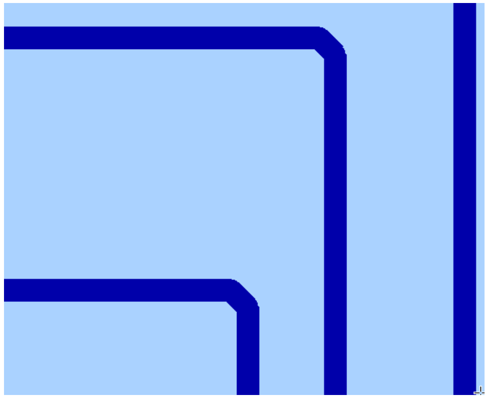
Note the miters are still pretty good and the max. allowed is achieved except for one.

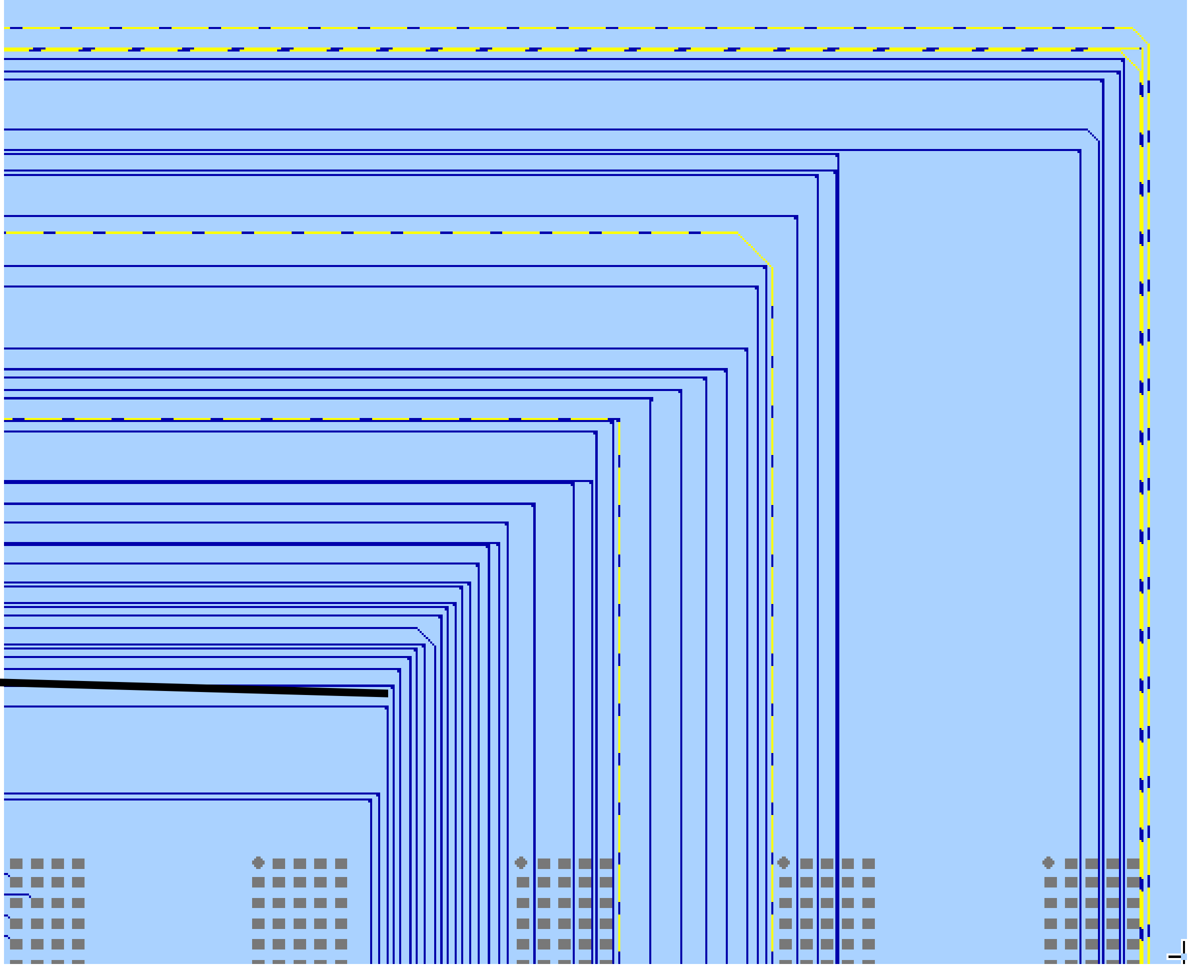
Miter not mitering to meet max allowed

International Cadence Usergroup

2. circuit selected (length 1.05   0.7 (type ratio))

Note the miters are NOT good and the max. allowed is NOT achieved.





Miter not mitering to meet max allowed

- **Miter not mitering to meet max allowed, when easy min. rule exits as well as max.**
  - **This was isolated by running 3 tests on the same design and do file except:**
  - **If same rules applied to all 3 (after in case of 1.).**

1. **No length rules at all** **Conflicts 284**
2. **Max. allowed length limited** **Conflicts 298**
   circuit selected (length 1.05 (type ratio))
3. **Max. allowed length limited, plus easy min. required** **Conflicts 345**
   circuit selected (length 1.05 0.7 (type ratio))

Miter not mitering
What to do??

International
Cadence
Usergroup

- – Turn off the min. rules
- – miter,
- – then turn on the min. length rules
- – and miter again to add delays OR
- – use the elongate command.

- Matched lengths rules are ultimately seen as max. & min. so they also need to be temporarily off to avoid this problem.
- Ditto for relative length rules.

Getting no unconnects and “as short as possible” etc.

- Making Specctra connect the unconnects.
- Identifying trapped unconnects early
- Cleaning the best result rather than the last one from a route series in spite of snowballing poly wires (planes).
- Making Specctra clear the last “few crossing conflicts”
- Making Specctra shorten up sloppy routing including hairpin loops. “wringer”
- (a set of do file sequences)

Getting no unconnects

International
Cadence
Usergroup

- Making Specctra connect the unconnects with a do file sequence that:
  - Identifies the unconnects creating a group of the fromto’s that are not connected.
  - Writes out a wire file containing the connected fromto’s
  - Deletes all wires
  - Selects and routes the unconnected fromto’s on the otherwise empty design.
  - Reads the previously routed wire on top
  - and runs route passes to clear the resulting conflicts.

Getting no unconnects

- **Identifying trapped unconnects early**
  - At the end of this an unconnect report shows those that will not route on an empty board, therefore must be stonewalled or trapped.

```
##DO FILE SEQUENCE
#try to connect unconnects & report trapped & stonewalled situations
vset system Unroute on
if (unconnect_wire > 0)
 then (
 report unconnects $/pre_try_connect.txt
mode select guide;        unsel all routing
#set co-ordinates to suit the units and full area of the design
select area guide 13110 -210 -360 12480
```

Getting no unconnects

```
define (group uncon__gr (selected))
unsel all routing;      mode measure
write wire $/pre_force_conn.w;      delete all wires
select group uncon__gr;      route 25 1;      clean 2
#workaround for poly duplicating bug
unsel all routing;select all poly_wires;     unprotect selected poly_wires
delete all poly_wires;                       unselect all poly_wires
read wire $/pre_force_conn.w
report unconnects $/post_try_connect_trapped_connections.txt
route 10 16
)
#end try to connect unconnects & report trapped & stonewalled situations
```

Getting best result rather than the last

- **Cleaning the best result rather than the last one from a route series in spite of snowballing poly wires (planes).**

```
bestsave on $/example_1_raw_best.w
#get best pass results
route 25 16;      delete all wires
select all poly_wires
unprotect selected poly_wires
delete all poly_wires; unselect all poly_wires
read wire $/example_1_raw_best.w
clean 2
#end get best pass results #repeat as required
```

Making Specctra clear the last “few crossing conflicts”

- Making Specctra clear the last “few crossing conflicts”
  - Toward the end of a complex design route specctra sometimes “stalls” or goes into a geometric loop and convergence is stopped.
  - Various interactive methods are used to change the view of things such as moving some things around and then running some more route passes.
  - The following do file sequence will do this automatically if the number of crosses is relatively low and distributed.
  - This do file sequence saves the wires at the start and will return to that status if the number of unconnects increases or crosses increase.
  - It monitors for reduction and stops when progress is not being achieved.

Making Specctra clear the last “few crossing conflicts”

International Cadence Usergroup

- **It deletes the crossed fromto’s, then runs cleans (which cannot put in conflicts) to change the view then reroutes the fromto’s that were in conflict.**

```
#start try to complete by clearing crosses,
set auto_remove off;       write wire $/tr1a_pre_wringer_start.w
if (conflict_crossing > 0)
 then (
setexpr improving_it (1);setexpr loop_it (0)
while (improving_it == 1  && loop_it < 10)
 (
write wire $/tr1a_checkpt.w
setexpr unconntr1a_start (unconnect_wire)
setexpr crossestr1a_start (conflict_crossing)
```

Making Specctra clear the last "few crossing conflicts"

```
delete conflicts;  clean 2;  tax cross 0.6;  route 5 16
tax cross 1.4;  route 25 16
setexpr unconntr1a_finish (unconnect_wire)
setexpr crossestr1a_finish (conflict_crossing)
if (unconntr1a_finish > unconntr1a_start)
  then (
sh echo not better;  delete all wires
select all poly_wires;  unprotect selected poly_wires
delete all poly_wires;  unselect all poly_wires
read wire $/tr1a_checkpt.w
setexpr improving_it (0)
       )
```

Making Specctra clear the last “few crossing conflicts”

```
if (unconntr1a_finish == unconntr1a_start && crossestr1a_finish >
    crossestr1a_start && loop_it > 1)
  then (
sh echo not better;  delete all wires
select all poly_wires;  unprotect selected poly_wires
delete all poly_wires;  unselect all poly_wires
read wire $/tr1a_checkpt.w;    setexpr improving_it (0)
      )
if (unconntr1a_finish == unconntr1a_start && crossestr1a_finish ==
    0)
  then (
setexpr improving_it (0)
      )
```

Making Specctra clear the last "few crossing conflicts"

```
if (unconntr1a_finish == 0 && crossestr1a_finish == 0)
  then (
setexpr improving_it (0)
       )
setexpr loop_it (loop_it + 1)
)
   )
#end try to complete by clearing crosses,
# keeping best result, not increasing unconnects
```

“as short as possible” “wringer” etc.

- Making Specctra shorten up sloppy routing including silly hairpin loops. “wringer”
  - This do file sequence sets temporary fromto length rules gradually tightening them up with routing passes at each stage. Starts with a Manhattan ratio of 3.0 decreasing gradually to 1.05 then gradually back up to 3.0.
  - At beginning it shortens any grossly long short Manhattans with some temporary actual length rules. The short ones are excluded from the main squeezing ratio sequence.
  - At the end it sets the temporary fromto length rules off with

```
circuit selected (length -1 -1 (type ratio))
circuit selected (length -1 -1 (type actual))
```

“as short as possible” “wringer” etc.

International Cadence Usergroup

- Making Specctra shorten up sloppy routing including silly hairpin loops. “wringer” (Cont. 1)
  - Has been used on hundreds of designs to:
  - Reduce total routed length by 10 to 20%
  - Reduce crosses and clearances
  - Reduce unconnects.
  - Reduces noise
- The length reduction frees up real estate which allows the unconnect and conflict reduction.
- The set auto_remove off command is essential to prevent Specctra panicking and ripping up a lot of routes.
- It does take a significant amount of time to run but always delivers worth while length reduction.

“as short as possible”
“wringer” etc.

International
Cadence
Usergroup

- Making Specctra shorten up sloppy routing including silly hairpin loops. “wringer” (Cont. 2)
- Because the conflicts are increasing initially during the tightening up, bestsave does not work so specific write wire file commands are used at each stage in case of a restart required.
- Key example sequence

```
select all nets
unselect fromto (length 0 1750)
circuit selected (length 1.5 (type ratio))
unselect all nets
route 5 26
clean 2
write wire $/checkpt.w
```

“as short as possible” “wringer” etc.

- Remember the miter test cases.

  Max. allowed length limited, plus easy min. required circuit selected (length 1.05   0.7 (type ratio))

  As run without clear crosses routine and wringer.

  Conflicts 359 (12 crosses, 0 clear, 2 Xtalk, 345 length)

  11808 routed length

  Same run with wringer and clear crosses

- Conflicts 196 (2 crosses, 0 clear, 2 Xtalk, 192 length)
- 10973 routed length
- The complete sequence is a bit long so it is included in the file
  - 10_do_file_fragments.txt (along with the others)

Bye

If no more questions then

ME