

0.1 はじめに

テープアウトに立ち上がるラスボス四天王は、Calibreで行うDRC/LVS/ERC/ANTである。それぞれのレイアウトツールで、これらのチェックをやった際は、大抵はうまく行ったはずだ。これしかやっていない人は、なんでみんながDRCやLVSで騒いでいるのか理解できないかもしれないが、レイアウトツール内のルールチェックは、いわば自分でやったテストの回答を自分で採点するようなもので、Violationがなくなっていればエラーがなくて当然なのである(それでもエラーはあるのだが、)。ところがテープアウトを行うためには、Calibreというツール上でこれらのチェックが4つ共、通らなければならない。

Calibreは、基本的にそれぞれのレイアウトツールが吐き出した図形データgdsを解析してチェックするので、自己チェックとは異なり、テープアウトに必要な、つまり製造に必要な厳しいチェックが図形レベルに行われることになる。こいつらをクリアするには最低2週間を確保すべきである。さらに、若くて徹夜ができ、相当な技量を有する「勇者」と「戦士」が必要で、勇者はDRCをクリアしてからANTへ進ませ、戦士にはLVS/ERCを担当させるのが良い。さらにテープアウト経験者でアドバイスをを行う「賢者」を置いて、煮詰まった場合のサポート体制を固めるのが良い。

ClubLayoutの皆様は、自分の設計で一度はこれを体験されたい。

0.2 ディレクトリセットアップ

四天王対応に、さのすけは、ふんが研内にMakefileを使ったチェック環境を確立している。例えば、hunga/verilog/f65/verifyにこの環境がある。で、実はふんがもこれがどのように動いているか、良く分かっていない。MakeとCalibreのバッチコマンドに詳しい方はぜひ解析を試みて下さいませ。

- scripts_icfb: calibre用のスクリプト、ディレクトリ毎コピーする。
- run: 実行ファイルを入れておく。
- input: チェック対象のgdsを置く。
- cdl: LVS用のcdlを置く。
- input内のgdsをシンボリックリンクを張って置いておく。
- Makefile: コピーして持ってくる。

0.3 セルとフレームのはめこみ

小林フローは、標準セルを使っておらず、京大のセルを使っているため、レイアウトツールから、全階層のデータを吐き出して、これにフレームを嵌める。ちなみにフレームは、レイアウトツールには供給されず、gdsのみが供給されるので、小林フローでも後で嵌め込むことになる。

ところが標準セルを使う場合、標準セルもフレーム同様に後から嵌め込む。標準セルは、中身が解放されているので、原理的には、小林フローと同じでもうまく行くんじゃないかと思うのだが、もしかして、レイアウトツールのセルとgdsで供給されるセルに違いがないとも限らないので、後から嵌め込む方が安全であるのは間違いない。

このため、ふんが研のフローは、すべてを単一階層で吐き出し、calibre上でこれらを嵌め込むようになっている。

```
make clean
```

```
make base
```

これで、セルライブラリ用の嵌め込み環境が用意される。これにマクロである POCOP.gds、トップ階層の POCOP_TOP.gds を嵌め込む場合、以下のように make する。

```
make POCOP.streamin
make POCOP_TOP.streamin
```

さらに、フレームをくっつけて、全体を吐き出すには以下のようにする。

```
make POCOP_TOP.addframe
```

Calibre が立ち上がるので、最終レイアウトを確認する。このコマンドの最後の方で、フレームの入った gds を input 内の POCOP_TOP.gds に書き潰す形で吐き出す。やや時間が掛かるので焦らないで待つこと。

レイアウトを修正した場合、修正した gds をそれぞれストリームインするのを忘れないこと。本当はそれでストリームインした部分だけが入れ替わるはずだが、ふんがコワがりなので、毎回最初からやっている。

1 DRC(Design Rule Check)

もっとも有名なデザインルールチェック。これが通らないと、LSI が動かないどころか、製造中に問題が起きるため、製造自体が許されない。DRC は、e-shuttle から供給されたスクリプトを実行してチェックする。やや分かりにくいのだが、スクリプトを生成するための実行ファイルが供給されている。これが、cal_drccs2001 である。これを富士通のディレクトリか hunga/verilog/f65/verify/run からでも verify の下の run にコピーする。

いきなり実行

```
./cal_drcc2001
```

やたら使い難いメニュー方式で聞いてくる。例えば、ストリームファイルの場所を設定したければ、1 を押すと、名前を聞いてくるので、パス付きで設定する。最終的に下の画面のように設定したら e を入れる。

```
* These are input files & fig. names list.
```

```
+-----+-----+-----+
| No. | Input files          |
+-----+-----+-----+
|  1  | Stream file          | ../input/POCOP_TOP.gds
|  2  | Stream top fig.      | POCOP_TOP
|  3  | Stream file system   | GDSII
+-----+-----+-----+
```

```
* Please enter Item No. to change.(input end ->'e/E')
```

次はチェック項目で、以下のように入れる。

```
* This is check items list.
```

```
+-----+-----+-----+
| No. | Item                  |
+-----+-----+-----+
|  1  | Technology            | CS200L
|  2  | Check mode            | NORMALCHK
|  3  | Geometry      check option | NOT_EXE
|  4  | Density      check option | WindowStep=1/1
```

```

| 5 | Power Gating check option | NORMAL
|   | (PGD_TEXT file name)      | not_necessary
+-----+-----+-----+

```

* Please enter Item No. to change.(input end ->'e/E')

次はプロセスについての情報で、下記のように設定する。

* This is process items list.

```

+-----+-----+-----+
| No. | Item          |
+-----+-----+-----+
| 1   | Well process  | TRIPLE_WELL
| 2   | Metal process | METAL_12_6321
| 3   | Power type    | P_12V_33V
+-----+-----+-----+

```

* Please enter Item No. to change.(input end ->'e/E')

今回はSRAMを使ってないので、これで良いと思う。

* This is process items list-2.

These inputs are used for USE-PROHIBITION MASKs check(Layout Rule).(*2)

```

+-----+-----+-----+
| No. | Item          |
+-----+-----+-----+
| 00  | 1.2V SVt -Tr.          use | YES
| 01  | 1.2V LVt -Tr.          use | YES
| 02  | 1.2V SRAM-Tr.          use | NO
| 03  | 1.2V RF Varactor        use | YES
+-----+-----+-----+
| 30  | 3.3V LVt -Tr.          use | YES
| 31  | 3.3V Poly-Pwell Cap.    use | NO
| 32  | 3.3V Poly-Nwell Cap.    use | NO
| 33  | 3.3V Sub-PNP Tr.        use | NO
| 34  | 3.3V RF Varactor        use | NO
+-----+-----+-----+
| 40  | I/O with Salicide Block use | YES
| 41  | N+ Silicide poly Res.   use | NO
| 42  | P+ Silicide poly Res.   use | NO
| 43  | P+ Non-silicide poly Res. use | NO
| 44  | MIM Cap.                 use | NO
| 46  | Laser Fuse               use | NO
| 47  | Electrical Fuse          use | NO
+-----+-----+-----+

```

*1 : If you select 'MIM15 Cap. = YES',
'MIM Cap.' automatically changes into 'NO'.

*2 : The design rules concerning each device layout

are always executed regardless of the option setting.

- * Please enter Item No. to change.(input end ->'e/E')
- * If you want ALL items change 'YES', please input 'ALL'.

次はデフォルトで良く、1を選択。

- * Please select the format of output file of DRC result.

- (1) GDSII & ASCII (*1)
- (2) GDSII & ASCII & RULE_DEBUG_GDS (*2)

- *1: ASCII error file name is 'drc_result.ascii'.
- *2: Output derivation layers for DRC (for rule development)

default(GDSII)

このまま実行する場合は1 そうでなければ no で終了。

- * Do you execute calibre job on this local machine,
yes(1) or no(default)?

後は、ファイル設定さえ変更しなければ、

POCOP_TOP_drc_run.csh

を実行すれば良い。毎回上記作業をやる必要はない。実行後、POCOP_TOP_drc.sum をチェックする。エラーがあれば、

make POCOP_TOP.open

とやると、Calibre とエラーブラウザが上がるので、ここで、

drc_result.ascii

を読み込んでやると、場所を表示してくれる。あとはこのエラーを取る方法だが、これはケースバイケースで大変である。これは、ぶっちさんの knowhow に譲りたい。

2 ANT(Antenna)

Antenna についての詳細な説明は、ぶっちさんの knowhow を参照のこと。アンテナも全て解決しないとテープアウトできないので、DRC をクリアした勇者は、次にアンテナに挑むべきである。

アンテナも DRC 同様に、

./cal_antcs2001

で、実行ファイルを生成する。

最初の設定は、DRC と同じである。

- * These are input files & fig. names list.

```
+-----+-----+-----+
| No. | Input files      |
```

```

+-----+-----+-----+-----+-----+-----+
| 1 | Stream file          | ../POCOP_TOP.gds
| 2 | Stream top fig.     | POCOP_TOP
| 3 | Stream file system  | GDSII
+-----+-----+-----+-----+-----+

```

* Please enter Item No. to change.(input end ->'e/E')

次は雰囲気が違うが、(8)を選択する。

* Please select number of metal structure(*1).

```

+-----+-----+-----+-----+-----+-----+
| No. | Metal option          | L L L L L L L | L L L L | L L | L | T |
|     |                       | A A A A A A A | B B B B | C C | D | M |
|     |                       | A B C D E F G | A B C D | A B | A | A |
+-----+-----+-----+-----+-----+-----+
| 1   | 07(5,0,1,1) layers   | * * * * *     |         | *   |   | * |
| 2   | 08(5,0,2,1) layers   | * * * * *     |         | * * |   | * |
| 3   | 08(5,1,1,1) layers   | * * * * *     | *       | *   |   | * |
| 4   | 08(5,0,1,1,1,) layers| * * * * *     |         | *   | * | * |
| 5   | 10(5,2,2,1) layers   | * * * * *     | * *     | * * |   | * |
| 7   | 12(5,4,2,1) layers   | * * * * *     | * * * * | * * |   | * |
| 8   | 12(6,3,2,1) layers   | * * * * * *   | * * *   | * * |   | * |
+-----+-----+-----+-----+-----+-----+

```

default : (8)

- *1 : In case of using c_mim and c_mimrf, 08(5,0,1,1,1) layers must not be selected.
- In case of not using inductor device, 08(5,0,1,1,1) layers must not be selected.
- In case of using inductor device, following layers must be selected.
- 08(5,0,2,1) layers
- 08(5,0,1,1,1,) layers
- 12(6,3,2,1) layers

で、これは終了で、あとはこのまま走らせる時は1 そうでなければnoで抜ける。

* Do you execute calibre job on this local machine,
yes(1) or no(default)?

同様に、

POCOP_TOP_ant_run.csh

が生成されるので、後は設定を変えなければこれを実行すれば良い。同様に、

POCOP_TOP_ant.sum

ができるので、これをチェックした上、

ant_result.ascii

で結果を検討する。

さて、アンテナ対策の基本はダイオードを付けることで、ICC は自動的にこれをやってくれるが、Astro は何もやってくれない。社長は、ガイドとブロッケージを駆使してアンテナの発生を抑えるフローを使っており、POCO, POCO_TOP でもこれを使っているが、完全に抑え込むことはできない。ダイオードなんて Verilog のソースに付ければ簡単じゃん、と思うと大間違いで、Astro は色々バッファなど入れてくれるため、入れたダイオードが機能しない場合が往々にしてある。アンテナの解決のノウハウはぶっちさんの knowhow を参照のこと。一度 Astro に Verilog を吐き出させてそこに付加してもう一度読ませるテクなどは圧巻で、アンテナテクに関しては彼の右に出るものは居ないのではないか、と、思う。

さらにちなみに ICC はアンテナ止めにダイオードを自動挿入してくれるのだが、これには落とし穴がある。これは ICC のフローを参照のこと。

3 LVS(Layout Versus Schematics)

3.1 LVS とは

LVS は、レイアウトツールから、Verilog ネットリストを吐き出させ、これを spice ネットリストに変換する。この spice ネットリストはシミュレーションには使えない LVS 専用のもので、cdl ファイルと呼ぶ。calibre は、gds から spice ネットリストを抽出し、この cdl と等価性をチェックする。これが等しければ、レイアウトツールからの Verilog が遅延付きシミュレーションが動作すれば、チップも必ず動くはずである。誤解しないで欲しいのだが、レイアウトツールに読み込ませた Verilog との等価性を検証するのではない。レイアウトツールはクロックツリーを作ったり、遅延調節のバッファを入れたり、Verilog ネットリストに様々な改変を行う。さらに I/O パッドやフィラーまで入れるので、これらを含めた出力との等価性をチェックするのが LVS である。

したがって、LVS が通って、レイアウト後のシミュレーションが動けば、多分、そのチップは動くはずである。今まで LVS が通って動作がおかしかったのは Geysler-2 だけである。

しかし、gds から抽出したネットリストはそう簡単に Verilog から変換した cdl と一致するはずはなく、実際は、cdl を色々細工しなければならない。この作業は、DRC, ANT とは並行に行った方が良く、さらに、LVS が通らなくても最悪の場合、言い訳を書くことで自己責任でテープアウトすることが可能である。ということで、この LVS とこれとセットになっている ERC は、メイン設計者である勇者ではなく、サブ設計者の戦士に任せるべきである。ただし、言い訳してテープアウトした SMA-1 は FIB 加工が必要になったことを考えると、LVS を通さないでテープアウトをすることは、パーティが過労で全滅するまで避けるべきである。(SMA-1 の際は実際全滅したので仕方なかった)

3.2 cdl の生成と ed ファイルの作成

input に、POCOP_lvs.v, POCOP_TOP_lvs.v を用意しておく。この Verilog ファイルは、電源、グランドピンを含めて単一階層で出力したものである。

verify ディレクトリで、以下を実行する。

```
make POCOP.cdl
```

```
make POCOP_TOP.cdl
```

これで cdl ファイルが cdl ディレクトリの下に生成される。cdl ディレクトリの中には、あらかじめ、それぞれの部品の cdl ファイルを置いておき、make 時にこれを include するようになっている。

cs202_fm.cdl, cs2020_io.cdl, cs202pg_uc_ail.cdl, CORNER_WIRE.cdl, VPW_VSS.cdl, ps_filler.cdl である。これはあらかじめコピーして置いておく。

次に、トップ階層についてのみ以下の細工をする。

I/O パッド、I/O ワイヤパッド、I/O フィラー、コーナーセルについて

を削除する。

最後に、.GLOBAL VDE を付け加える。また.INCLUDE にマクロの POCOP.cdl を付け加える。

ちなみに、コーナーセルを除いて IOCB2 という部品名を持っているので、これを利用して、削除を自動で行ってくれるのが社長の書いた ruby で、rm_well.rb であるが、遅いので注意。C で書き直そうと思っているのだが、やる暇がない。誰かやって。

これで Astro の場合は OK である。しかし ICC はまだ細工が必要で、これについては、ICC のフローを参照のこと。これは ICC のみ変だと思うが、往々にして、cdl は改変の必要があるので、フレキシブルにやる。とはいえ、あんまりこれを相手方に合わせて改変すると何をやっているのかわからなくなってしまうけど。

次に ed ファイルを作る。これは calibre が入出力ピンの TEXT を認識できないため、これを与えてやるファイルである。これは入出力のパッドの位置をめぐらして、テキストファイルを定義するもので、本当に座標を指定しなければならない。非常に面倒なのだが、幸いにしてパッドは大きいので、多少ずれても大丈夫である。色々な人が、ピン配置表からこれを自動生成するツールを作っているのだが、どれも ad hoc なので、伝わっていない。小林先生は非常にちゃんとしたのを作っているが、Rohm 用で使えない。誰か決定判を作ってくれるとうれしい。

```
LAYOUT TEXT "VDE" -1710 -908 617 POCOP_TOP
LAYOUT TEXT "VSS" -1650 -908 617 POCOP_TOP
LAYOUT TEXT "VDD" -1590 -908 617 POCOP_TOP
LAYOUT TEXT "VDE" -1530 -908 617 POCOP_TOP
LAYOUT TEXT "VSS" -1470 -908 617 POCOP_TOP
LAYOUT TEXT "VDD" -1410 -908 617 POCOP_TOP
LAYOUT TEXT "VDE" -1350 -908 617 POCOP_TOP
LAYOUT TEXT "VSS" -1290 -908 617 POCOP_TOP
LAYOUT TEXT "VDE" -1230 -908 617 POCOP_TOP
```

```
LAYOUT TEXT "IO_WE_N" -1170 -908 617 POCOP_TOP
LAYOUT TEXT "IO_DADDR[7]" -1110 -908 617 POCOP_TOP
LAYOUT TEXT "IO_DADDR[6]" -1050 -908 617 POCOP_TOP
LAYOUT TEXT "IO_DADDR[5]" -990 -908 617 POCOP_TOP
LAYOUT TEXT "VSS" -930 -908 617 POCOP_TOP
LAYOUT TEXT "VDD" -870 -908 617 POCOP_TOP
LAYOUT TEXT "IO_DADDR[4]" -810 -908 617 POCOP_TOP
LAYOUT TEXT "IO_DADDR[3]" -750 -908 617 POCOP_TOP
LAYOUT TEXT "IO_DADDR[2]" -690 -908 617 POCOP_TOP
LAYOUT TEXT "IO_DADDR[1]" -630 -908 617 POCOP_TOP
LAYOUT TEXT "IO_DADDR[0]" -570 -908 617 POCOP_TOP
LAYOUT TEXT "IO_DDATAOUT[15]" -510 -908 617 POCOP_TOP
LAYOUT TEXT "IO_DDATAOUT[14]" -450 -908 617 POCOP_TOP
LAYOUT TEXT "IO_DDATAOUT[13]" -390 -908 617 POCOP_TOP
LAYOUT TEXT "VSS" -330 -908 617 POCOP_TOP
LAYOUT TEXT "VDD" -270 -908 617 POCOP_TOP
```

...

3.3 実行

DRC, ANT と同様に、

./cal_lvscs2001

を実行。LVS と ERC は一度に実行できる。チップレベルで最終結果を求めるならば、1 を選ぶ。マクロを ed ファイルなしで ERC だけやっとく際は 2 を選ぶ。

* Please select check mode.

- (1) LVS (&ERC) (GDS, CDL, EDTEXT, CellMap, B-Box files)
- (2) ERC (GDS, EDTEXT files)

チップを検証する際は 1 を選ぶ。

* Please select check object.

- (1) CHIP (*1)
- (2) NORMAL(MACRO,HLB)

*1:EXCLUDE CELL 'WIMMLX_CS200L, WIMMLX_{UR, UL, DL and DR}'
default : (1)

ファイル名等をセットする。7 では、先ほど作った ed ファイルを指定する。9 はメモリを使う際は指定する。

* These are input files & fig. names list.

No.	Input files	
1	Stream file	../input/POCOP_TOP.gds
2	Stream top fig.	POCOP_TOP
3	Stream file system	GDSII
4	Schema file	../cd1/POCOP_TOP.cdl
5	Schema top fig.	POCOP_TOP
6	Schema file system	CDL

No.	Lib. files	
7	EDTEXT file	MANUAL input
	(file name)	POCOP_TOP.ed
8	CellMap file	AUTO input
	(file name)	not_necessary
9	B-Box file	NOTHING input
	(file name)	not_necessary

* Please enter Item No. to change.(input end ->'e/E')

プロセスはいつものを指定する。

* This is process items list.

No.	Item	
-----	------	--

| | | Shema (YES) AF K

+-----+

*1: Metal terminal cut puseudo resistor

* Please enter Item No. to change.(input end ->'e/E')

次も良くわかんないのだが、以下のように設定

* This is check tolerance value list.

+-----+

| No. | Item | value ('x/X': no check)

+-----+

| 1 | Tr. gate length [%] | 1

| 2 | Tr. gate width [%] | 1

| 3 | Poly RES. value [%] | 1

| 4 | Poly RES. width [%] | x

| 5 | CAP. value [%] | 1

| 6 | ESD DIODE perimeter [%] | 1

| 7 | ESD DIODE area [%] | 1

| 8 | I/O RES. width [%] | 1

| 9 | I/O RES. length [%] | 1

+-----+

* Please enter Item No. to change.(input end ->'e/E')

次は ERC のチェック項目だが、アナログダイオード以外は全部やしないとダメ。

* This is ERC check items list.

+-----+

| No. | Item |

+-----+

| 1 | Well connect check (ERC) | YES

| 2 | Tr. p-g short check 1 (ERC) | YES

| 3 | Tr. p-g short check 2 (ERC) | YES

| 4 | Floating Node(PATHC1) (ERC) | YES

| 5 | Floating Node(PATHC2) (ERC) | YES

| 6 | Floating Node(PATHC3) (ERC) | YES

| 7 | Floating Node(PATHC4) (ERC) | YES

| 8 | Latch Up check (ERC) | YES

| 9 | Analog diode check (ERC) | NO

+-----+

* Please enter Item No. to change.(input end ->'e/E')

次は、1のみで良い。

* Please select the format of output file of result.

(1) ASCII

(2) ASCII & MASK SVDB DIRECTORY

最後は、このままやるならば1、そうでなければno

```
* Do you execute calibre job on this local machine,  
yes(1) or no(default)?
```

一度できてしまえば、

```
./POCOP_TOP_lvs_run.csh
```

で実行可能。結果は、

```
POCOP_TOP_lvs.sum
```

に表示される。

3.4 結果の解析

まず、cdl内には空のSUBCKT文が混ざっているため、以下のようなワーニングが出るがこれは気にしなくていい。

```
Warning: Duplicate subckt definition "ZCGCB2E4COXA1" at line 1 in file "../cdl/././CORNER_WIRE.cdl"
```

しかし、ここにErrorが出ると、等価性比較自体をしてくれない。NOT COMPAREDという最悪の結果になる。これは対応するモジュールのピンがおかしい、SUBCKTが存在しない場合になる。

比較自体をしてきて、失敗すると、INCORRECT表示が出る。どこが違ったかを表示してくれるのでこれを見て当たりを付けるのだが、正直言ってあっさり、お手上げになることが多い。攻め口は以下の通り

- それぞれのモジュール単位でCORRECTかどうかの結果がでる。ここで、I/Oセルや標準セルなどでINCORRECTが出ることもある。本来セル内で出るのははずがないのだが、これが出るということは、このセルを使っている辺が怪しいことになる。
- LVSは実行中、gdsから生成したcdlをPOCOP_TOP.layout.net.gzに吐き出す。これを見ながら目で比較する。しかし、階層構造が全く違うので苦戦することが多い。
- ERCをしてみる。両者は関連している時があるので、ERCがごっちゃり出る辺に不一致があるかもしれない。

で、最終的にうまく行くと、LSI設計者の中で有名な「にっこり君」が表示される。

```
          #          #####          - -  
          #          #          #          * *  
    #    #          #    CORRECT    #          |  
    #    #          #          #          \___/  
    #          #####
```

「にっこり君」は、戦士にとって勝利と希望の象徴で、これを見ると涙せざるを得ない。しかし、TEDでサポートするツールでは、これが出てこない。ふんがは、TEDの担当に、にっこり君がでないLVSは信用できんと騒いだため、相手は大変困っていた。

4 ERC(Electric Rule Check)

これは、電氣的にルールを満足しているかを調べるもので、LVSと同時に実行できる。マクロでedファイルを作るのが面倒な場合、ERCだけを実行するのも手である。結果は、

```
POCOP_TOP_erc.sum
```

中に表われる。これにはあらかじめ VDEC 公認の擬似エラーが I/O バッド上に表われるがこれは気にしないでもいい。

```
--- ERC RULECHECK RESULTS STATISTICS (BY CELL)
---
CELL IOCB2EPG5PB11 ..... TOTAL Result Count = 433 (15588)
    ERC CHECK ERC_pgshort_NMOS:1 ... TOTAL Result Count = 200 (7200)
    ERC CHECK ERC_pgshort_NMOS:2 ... TOTAL Result Count = 200 (7200)
    ERC CHECK ERC_pathchk:2 ..... TOTAL Result Count = 33 (1188)
CELL IOCB2EPE5PE11 ..... TOTAL Result Count = 510 (9690)
    ERC CHECK ERC_pgshort_NMOS:1 ... TOTAL Result Count = 246 (4674)
    ERC CHECK ERC_pgshort_NMOS:2 ... TOTAL Result Count = 246 (4674)
    ERC CHECK ERC_pathchk:2 ..... TOTAL Result Count = 18 (342)
CELL IOCB2EPD5PI11 ..... TOTAL Result Count = 819 (17199)
    ERC CHECK ERC_pgshort_NMOS:1 ... TOTAL Result Count = 402 (8442)
    ERC CHECK ERC_pgshort_NMOS:2 ... TOTAL Result Count = 402 (8442)
    ERC CHECK ERC_pathchk:2 ..... TOTAL Result Count = 15 (315)
```

他にも宇佐美研製作のレベルシフトなどでは、エラーが出るが、これは擬似エラーとして申請して良い。

YUZYUBA をちゃんと入れないと、patchk:3, patchke:4 が出るが、これは擬似エラーとしてはダメで、ちゃんと入れてやる必要がある。

場所は、POCOP_TOP_erc.ascii を読み込むと calibre 上で特定できる。ただ、これはもちろん、擬似エラーも表示される。

5 最後の大ボス

四天王を倒しても、実は最後の大ボスが居る。これは VDEC の web 経由で、再び四天王 + α と戦わなければならない。(この辺、テープアウトは本当に RPG と似ている。) いくつか覚えておいておく。

- フレーム嵌め込みを間違えると GDR エラーがでる。これで TED と揉めたが、ふんがは向こうのミスと信じている。
- USE_xxV_Res_IOSB という項目をチェックしないと DRC エラーが I/O 上に発生する。これは SMA-1 の時池田さんに叱られた。
- RAM を使う場合、ブラックボックス指定ではなく、.INCLUDE "RAM196.cdl"などを付ける。
- YUZS のみで YUZYUBA 間を埋めると FL 密度チェックに引っかかる。YUZYB を混ぜるか、YUZYB だけでやれば大丈夫。これは Geysler-2 の時にはまった。