

デザインコンテスト・プレーヤ作成の落とし穴

正確なルールは の通りですが、

<http://lut.eee.u-ryukyu.ac.jp/dc13/rules.html>

"first player", "second player" の扱いが落とし穴になっていたようで、

よく(普通に?)読むと、流れとしては、ホストの動作はこんな感じなのですが、

1. ホストシステムは、接続された二つのクライアントに対して、生きてるか確認するために '0' を送ってくる
2. ホストシステムは、接続された二つのクライアントに対して、"first player" あるいは "second player" かを決める
3. ホストは、"first player" には、(5,5)に何か置けというコマンドを、"second player" には、(A,A)に何か置けというコマンドを送る
4. ホストは、"first player" と "second player" のどちらを先手にするか決めて、(これは 'r' オプションで決まるらしい)先手に決めた方に、3XXXX というコマンドを送る。
5. "first player" と "second player" の後手に決めた方に、4XXXXYYYY というコマンドを送る。
6. 以降は、ゲームがおわるまで、3XXXX というコマンドが送られる。

ここで、落とし穴は、(2) のところで (A,A) を受信した時点で

「自分は "second player" だから後手」だ、と、早とちりしてしまうと、

3XXXX が送られてきたときにエラーとしてしまっただけで変なところにおくか、

4XXXXYYYY がくるまで待ち続けてしまう、ということで無効になる、と。

デザインコンテスト

残念ながら(当然ながら?)予選落ち。

某高位合成処理で実装したところにも負けちゃったのが残念。

12月を目指してがんばれる...か?

話を聞いてると、上位のチームは、

忙しい中でもちゃんと時間を割いていらっしゃるようなので、

出るからには、もっとがんばろう。

FPGAで実装する

からには、パイプライン処理でリソース使用率を向上させることこそが鍵なんだろうなあ、とあらためて思う、など。

リソースシェアリングの究極が(おそらく)CPUなわけで、

そういう風の実装できるように

- ・アプリケーションを選ぶ
- ・アルゴリズムを工夫する
- ・がんばって記述する
- ・あるいは、手を抜いても、その観点でちゃんとできる言語で楽しく書く

とかが、大事なんだろうなあ。