

Verilogと戯れる

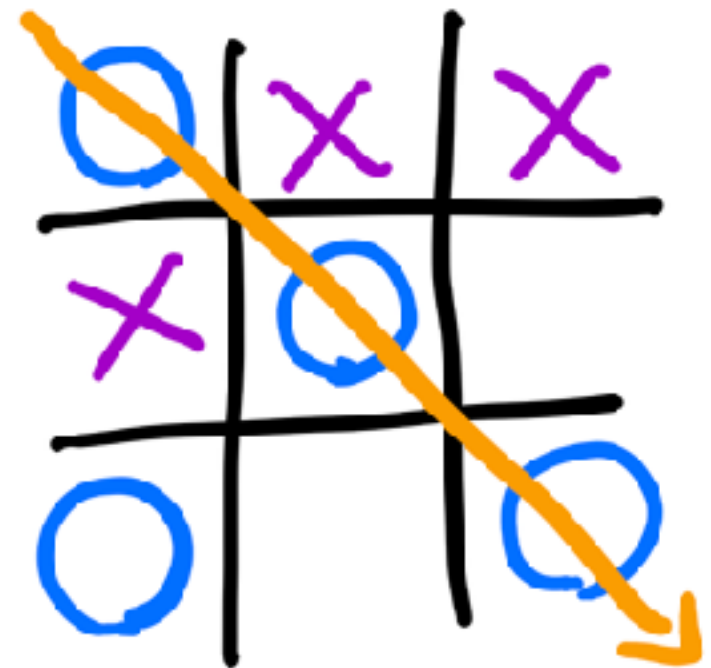
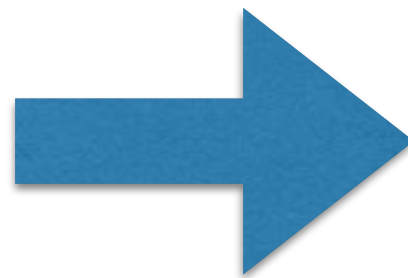
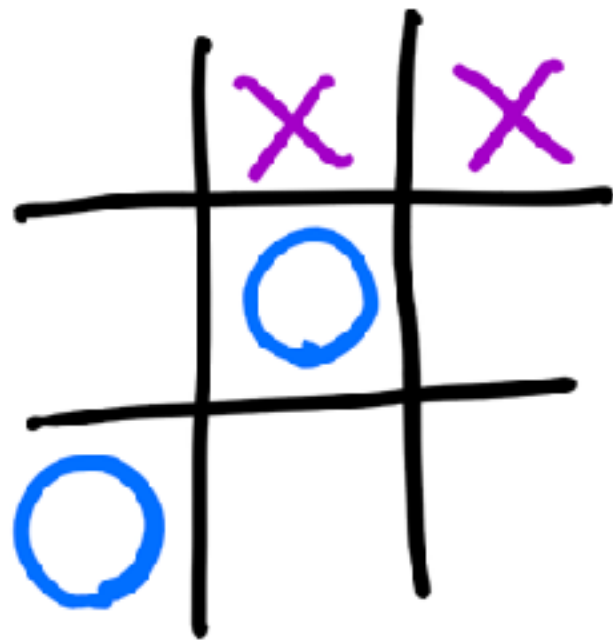
00-640726 桂 宏行

つくったもの

- ・ Tic Tac Game
- ・ （それに付随してVerilogを補助するパーサ）

Tic Tac Toe

- ・ 3x3のマス目に交互に丸ばつを入れていくゲーム
- ・ 縦横斜めどれかに一列並べられた方が勝ち



実装した内容

- ・ ゲーム実装
 - ・ 盤面表示
 - ・ 勝敗判定
- ・ 対戦CPU
 - ・ 乱択（っぽいAI）
 - ・ 最善手を打ってくるAI
- ・ その他ゲームらしい機能
 - ・ ゲームクリア～って出たりとか

AIについて

- ・ 盤面が3x3なので 3^9 で全探索可能
- ・ 引き分け以上を狙うAIを深さ優先探索で書いた
- ・ (再帰とか回路ではできないのでちょっとめんどくさかった)

大変だったポイント

- ・ ハードウェア的な問題
- ・ デバッグの問題
- ・ Verilogを書くのがきつい問題

ハードウェア的な問題

ハードウェア的な問題

次のようなコードを考えます

ハードウェア的な問題

次のよう

```
2 x は2bitの値
3
4 if (x == 2'b00) begin
5     黄色の円を表示
6 end
7 else if (x == 2'b01) begin
8     緑色の円を表示
9 end
10 else if (x == 2'b10) begin
11     青色の円を表示
12 end
13 else if (x == 2'b11) begin
14     白色の円を表示
15 end
16 else begin
17     赤色の円を表示
18 end
```

ハードウェア的な問題

次のよう

```
2 x は2bitの値
3
4 if (x == 2'b00) begin
5     黄色の円を表示
6 end
7 else if (x == 2'b01) begin
8     緑色の円を表示
9 end
10 else if (x == 2'b10) begin
11     青色の円を表示
12 end
13 else if (x == 2'b11) begin
14     白色の円を表示
15 end
16 else begin
17     赤色の
18 end
```

これが起こる。

のelseって
ない？

ハードウェア的な問題

次のよう

```
2 x は2bitの値
3
4 if (x == 2'b00) begin
5     黄色の円を表示
6 end
7 else if (x == 2'b01) begin
8     緑色の円を表示
9 end
10 else if (x == 2'b10) begin
11     青色の円を表示
12 end
13 else if (x == 2'b11) begin
14     白色の円を表示
15 end
16 else begin
17     赤色の
18 end
```

これが起こる。

のelseって
ない？

ハードウェア的な問題

次のよう

```
2 x は2bitの値
3
4 if (x == 2'b00) begin
```

驚くべきは、if文の中身が（挙動上）中途半端に実行されているようだったこと

```
12 end
13 else if (x == 2'b11) begin
14     白色の円を表示
15 end
16 else begin
17     赤色の
18 end
```

これが起こる。

のelseって
ない？

ハードウェア的な問題

- ・ 理由は、ボタン入力のチャタリングのため
- ・ チャタリング除去をすると治った

デバッグの問題

デバッグの問題

- ・ 基本的に、Verilogでコードを書き、
- ・ FPGAに乗せて挙動を確認

デバッグの問題

- ・ 基本的に、Verilogでコードを書き、

- ・ FPGAに乗せて挙動を確認
この繰り返しで実装

デバッグの問題

- ・ Pythonで動くコードをVerilogに焼き直したら動かない＞＜
- ・ しかも、どのパラメタが壊れているのかわからない

デバッグの問題

- ・ printfデバッグがしたい
- ・ が、printfは当然できないので、

デバッグの問題

- ・ printfデバッグがしたい

- ・ が

```
if (param < 0 || param > 10) begin  
    画面を赤くする  
end
```

- ・ このようにおかしい値を検知して気合いで直していた

デバッグの問題

- ・ printfデバッグがしたい

- ・ が

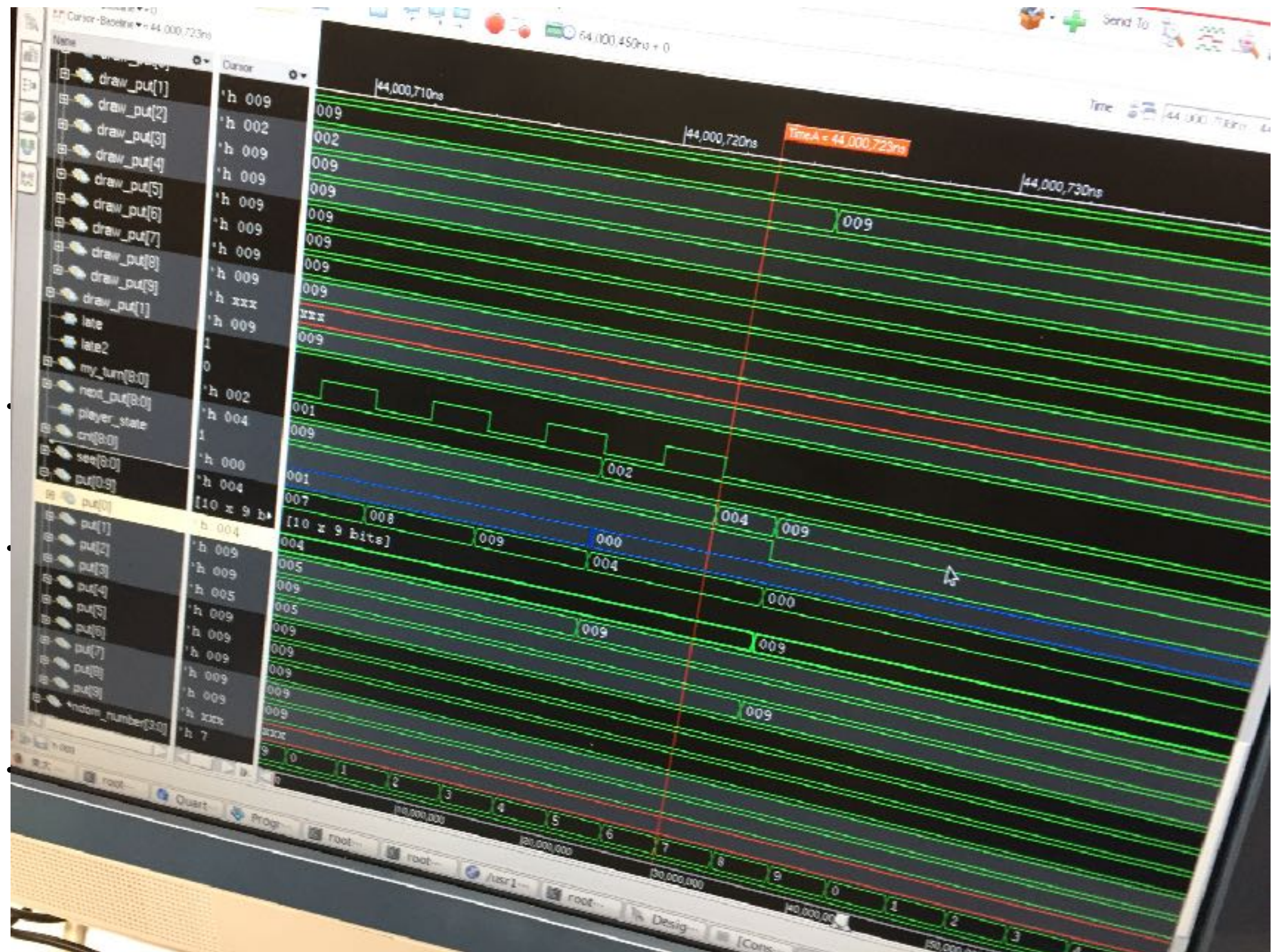
```
if (param < 0 || param > 10) begin  
    画面を赤くする  
end
```

- ・ このようにおかしい値
た

しかし、深さ優先探索の部分のコードが
どうも動かない

デバッグの問題

- ・ シミュレータを変数の値を可視化して解決
- ・ (TAの方にシミュレータを実行するコードを書い
てもらった)
- ・ これにより実質的なprintfデバッグができた



特定のクロックでの
レジスタの値がわかる



Verilogを書くのがきつい問題

Verilogを書くのがきつい問題

回路に組み込むときには、いくつか制限があり、以下の欲求があった

- ・ forは使いたい
- ・ 定数を1箇所に管理したい
- ・ コンパイル前に解決できる定数を解決したい

Verilogを書くのがきつい問題

回路に組み込むときには、いくつか制限があり、以下の欲求がある

勢いで言語拡張を書いていた

- ・ forは使いたくない
- ・ 定数を1箇所に管理したい
- ・ コンパイル前に解決できる定数を解決したい

Verilogを書くのがきつい問題

実装した機能

- ・ forによるループ展開
- ・ 定数の埋め込みと内部でPythonによるスニペット実行
- ・ importによるファイル分割
- ・ 特定の場所に画像を埋め込む機能

Verilogを書くのがきつい問題

- ・ forによるループ展開

回路を書く時にforは非推奨らしい（展開してくれるコンパイラもあるようだが、そうでないものも存在するため）

するとコピペコーディングをしまくることになる

これはつらい

Verilogを書くのがきつい問題

- ・ forによるループ展開

回路を書く時にforは非推奨らしい（展開してくれるコンパイラもあるようだが、そうでないものも存在するため）

するとコピペコーディング

forによる展開を自動生成
できるようにした

これはつらい

- ・ forによるループ展開

具体的には、次のような感じ

```
[[for i in 1, 2, 3 {  
    if (x == {{i}}) begin  
        // hogehoge  
    end  
}]
```

・ forに

具体的には、次のような感

```
if (x == 1'd1) begin  
    // hogehoge  
end  
if (x == 2'd2) begin  
    // hogehoge  
end  
if (x == 2'd3) begin  
    // hogehoge  
end
```

```
// [Compile] 1502037502.728922
```

まとめ

- ・ VerilogでTic Tac Toeをするチップを設計した
- ・ 作ったものの自体は簡単だが、FPGA上で正しく動作させるのはソフトウェアで行うのと比べて難しかった