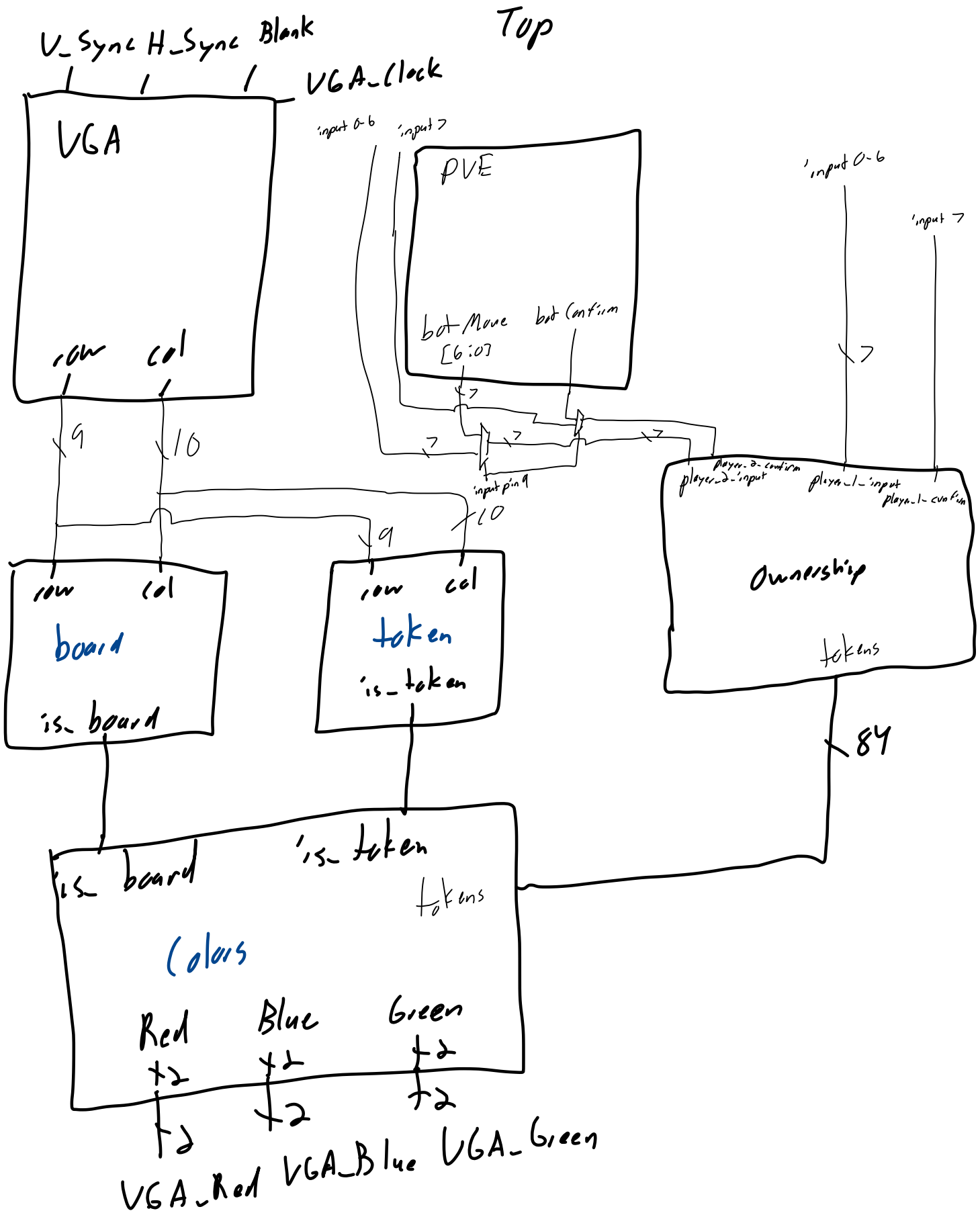
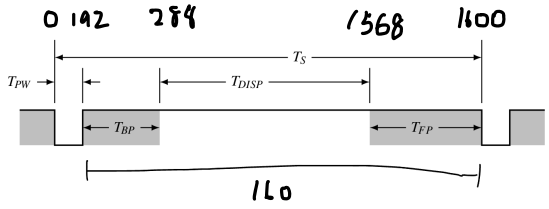

Top
V_Sync
H_Sync
Blank
VGA_Clock
VGA
row
col
9
10
input 0-6
input 7
PVE
bot Move
[6:0]
bot Confirm
7
input pin 9
10
9
row
col
board
is_board
row
col
token
is_token
input 0-6
input 7
7
player_2_confirm
player_2_input
player_1_input
player_1_confirm
Ownership
tokens
84
is_board
is_token
tokens
Colors
Red
Blue
Green
x2
x1
x2
x2
x2
x2
VGA_Red
VGA_Blue
VGA_Green

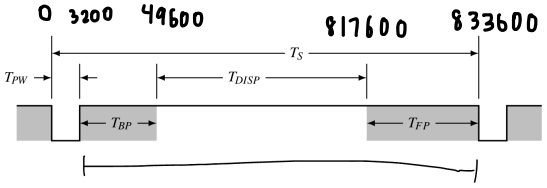
4 counters: 1 for row
1 for col
1 for vCount
1 for hCount

Following Design is for VGA module

Hcount:



Vcount:



Tdisp (VS) = 768000

T_s (HS) = 1600

768000 / 1600 = 480

There are 480 HS pulses during Tdisp for VS.

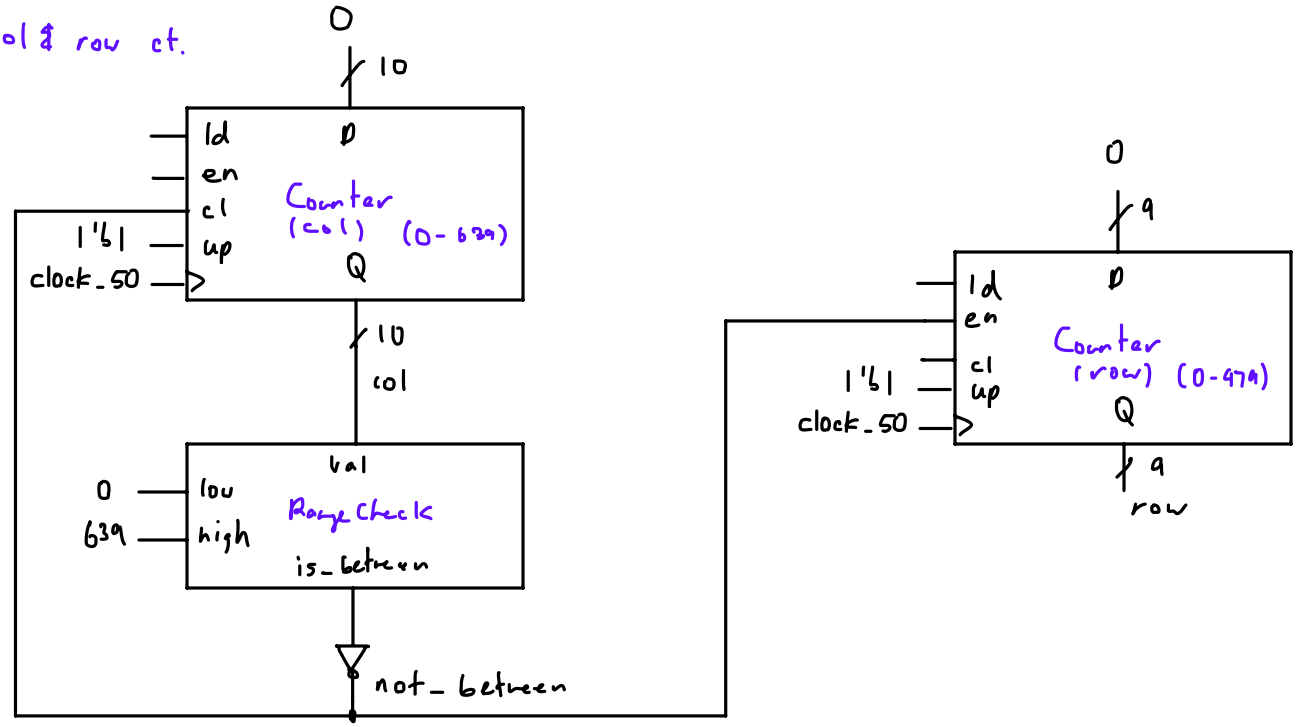
480 rows

7 strips of board
6 tokens
each token is 45 pixels tall
each strip of board is 30 pixels tall

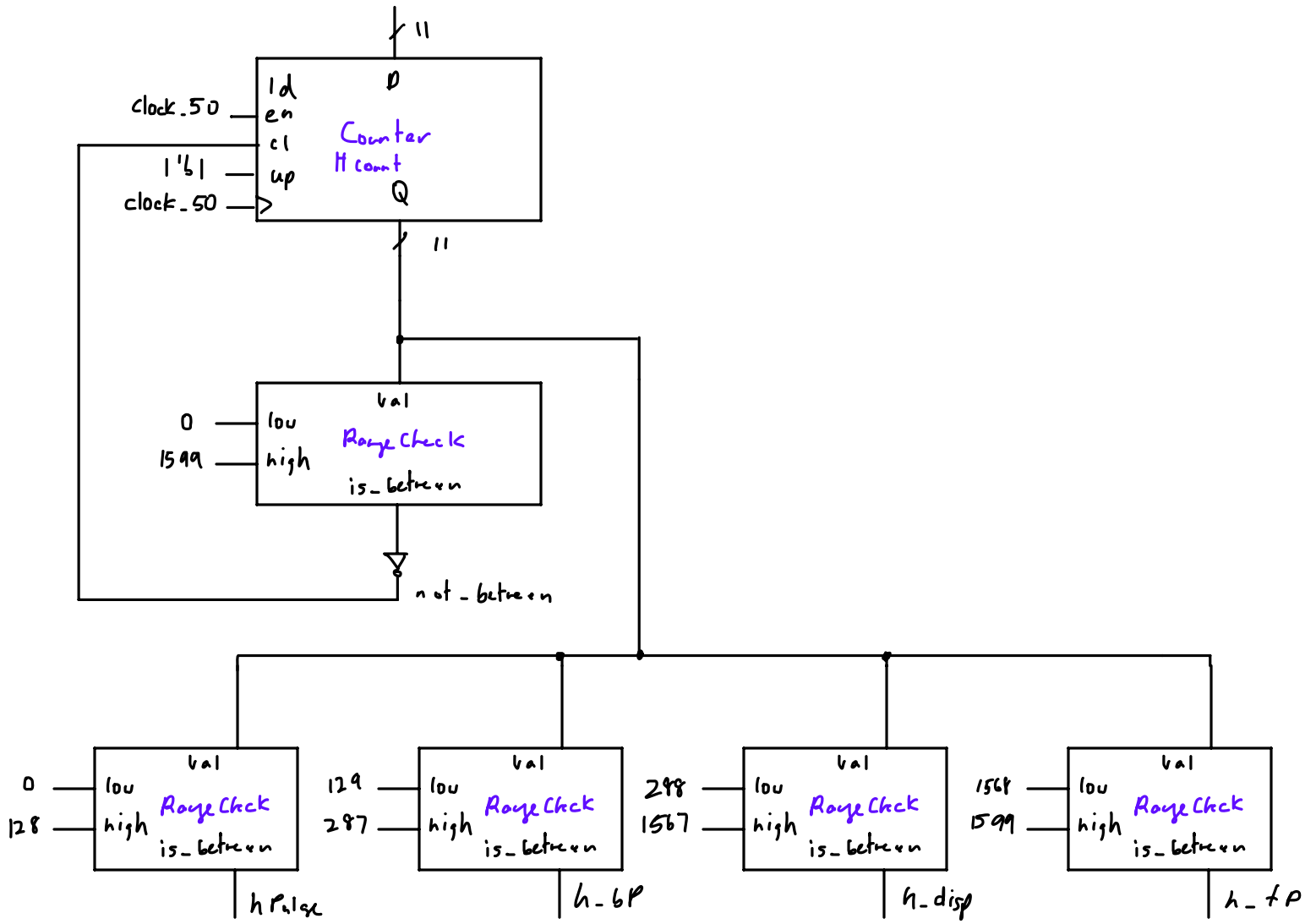
640 columns

7 tokens
8 strips of board
each token is 40 pixels wide
each strip of board is 45 pixels wide

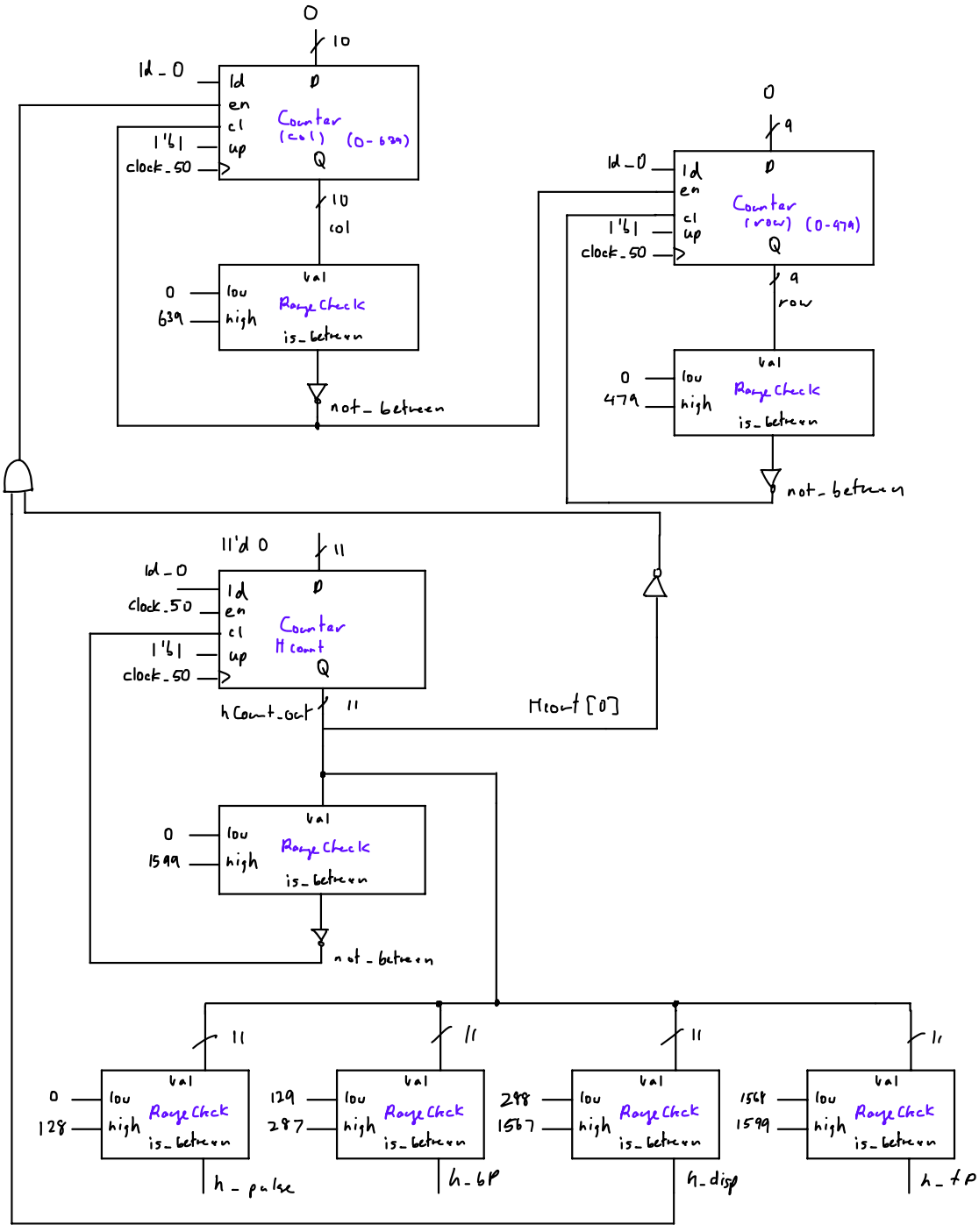
col & row ct.

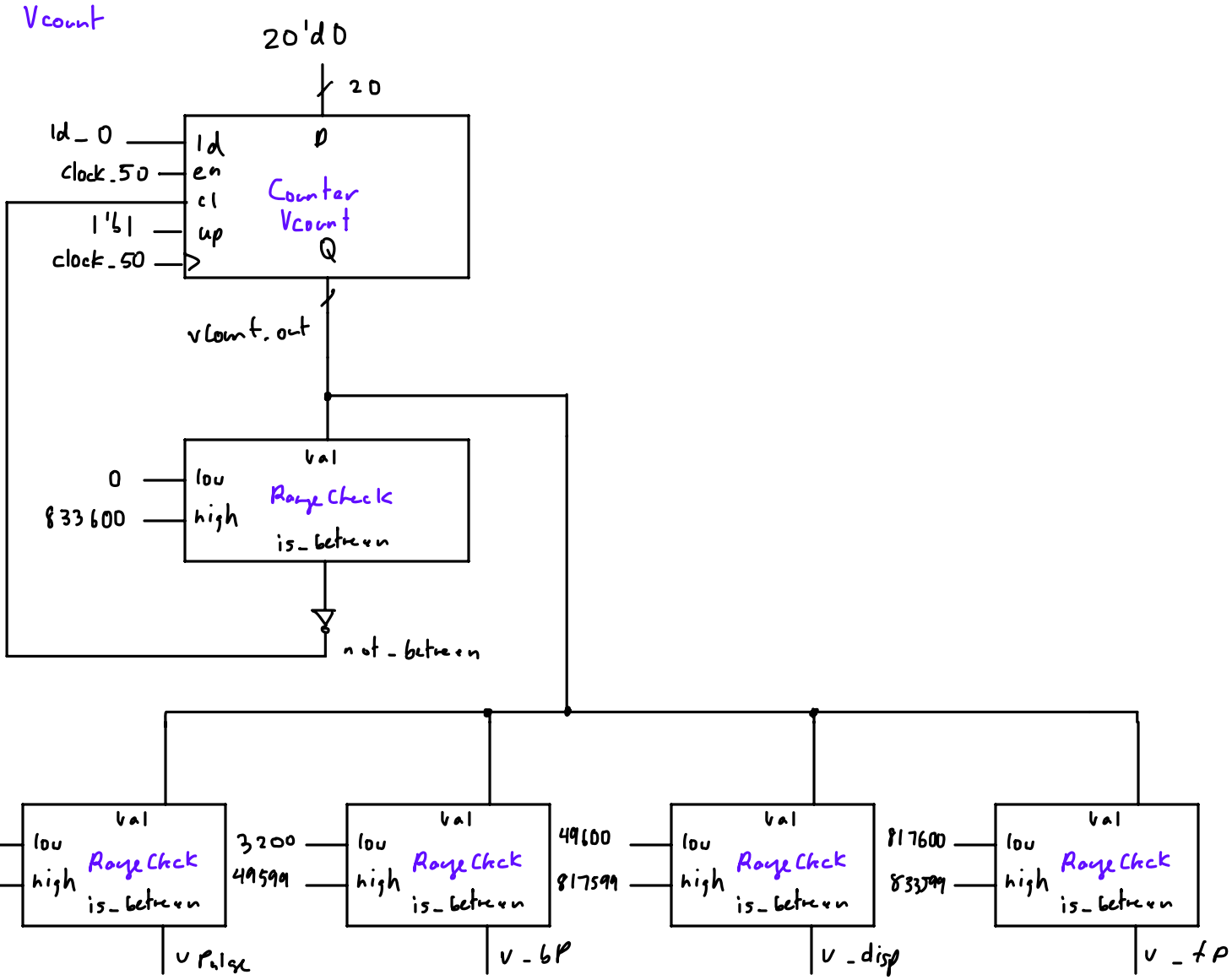


H count



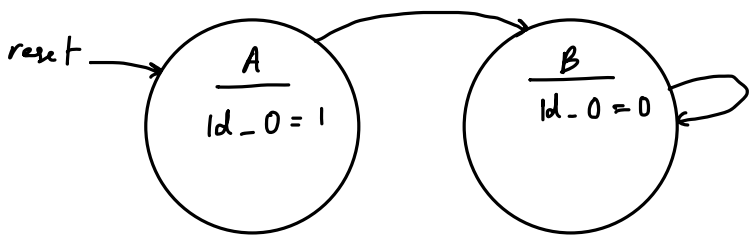
HCount & Row/Col count

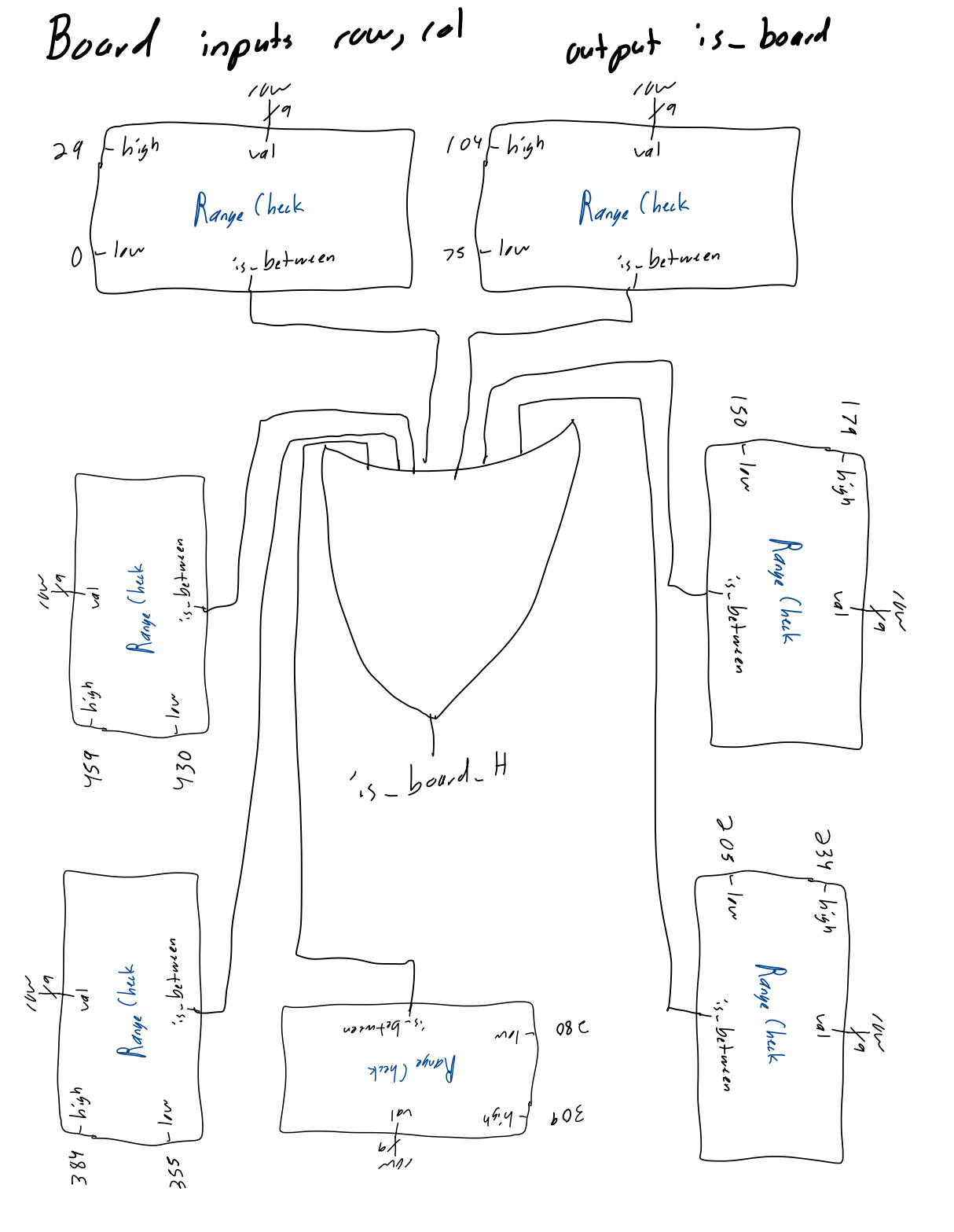




STD

Control points: ld_0




Board inputs row, col
output is_board
row
col
29
high
0
low
Range Check
is_between
104
75
459
430
384
355
309
380
150
174
205
234
is_board_H

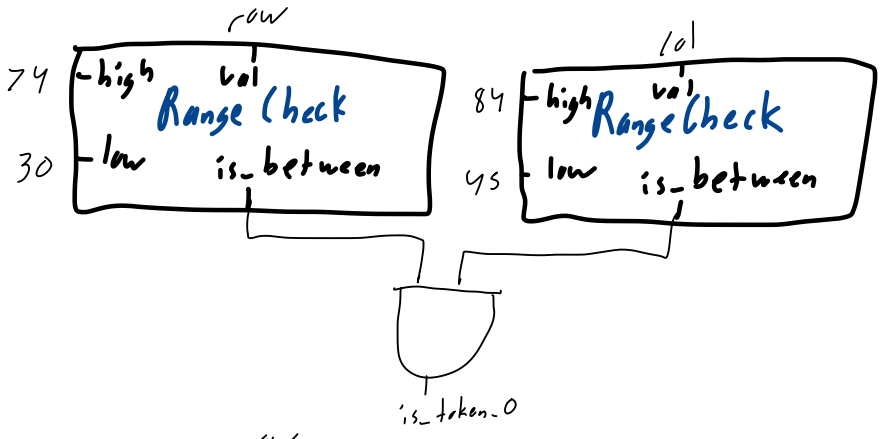
Board

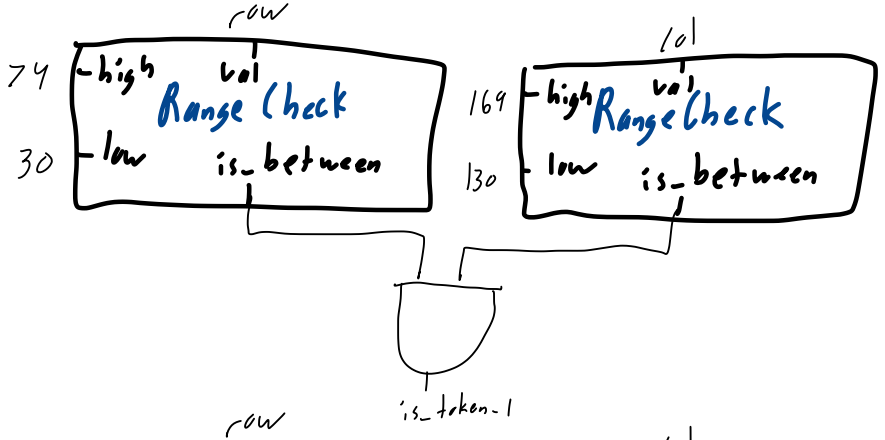
col
10
val
44
high
Range Check
0
low
is_between
col
10
val
129
high
Range Check
85
low
is_between
col
10
val
639
high
Range Check
595
low
is_between
col
10
val
554
high
Range Check
510
low
is_between
col
10
val
469
high
Range Check
425
low
is_between
col
10
val
384
high
Range Check
340
low
is_between
col
10
val
170
low
Range Check
214
high
is_between
col
10
val
255
low
Range Check
299
high
is_between
is_board_y

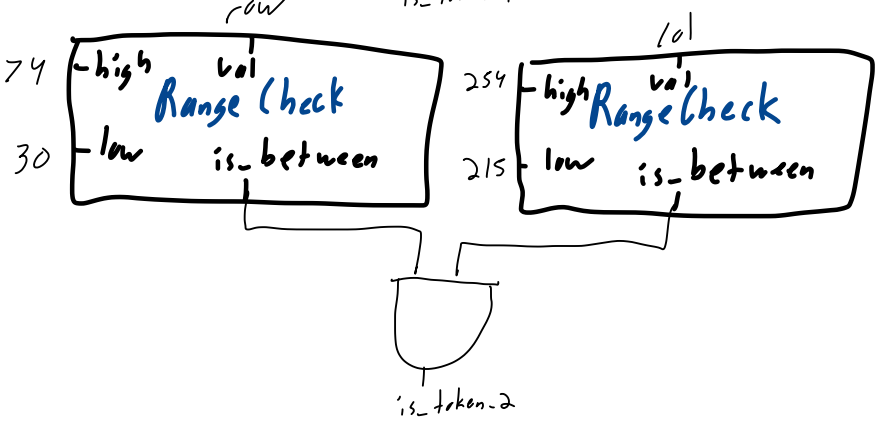
Board

```
is_board = is_board_V |
           is_board_H
```

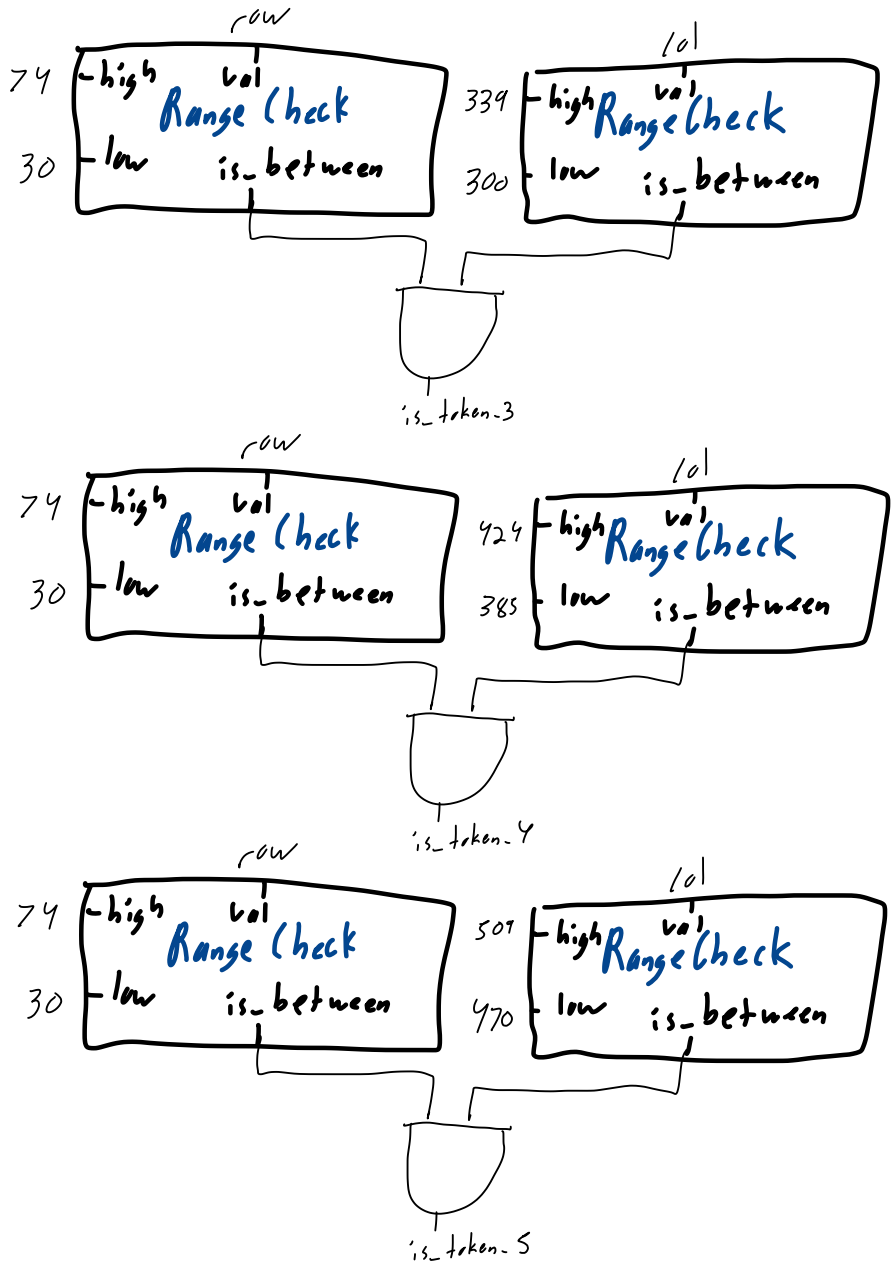
Token inputs: row, col output: is_token



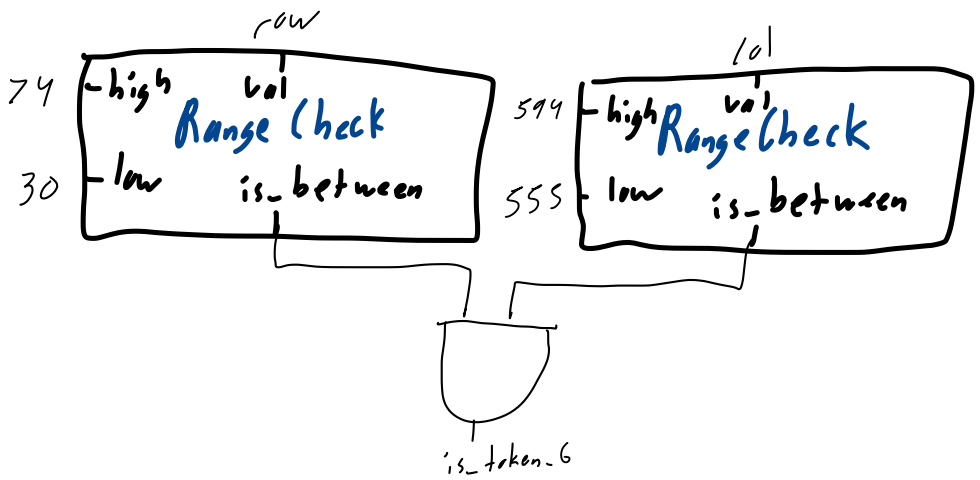




Token

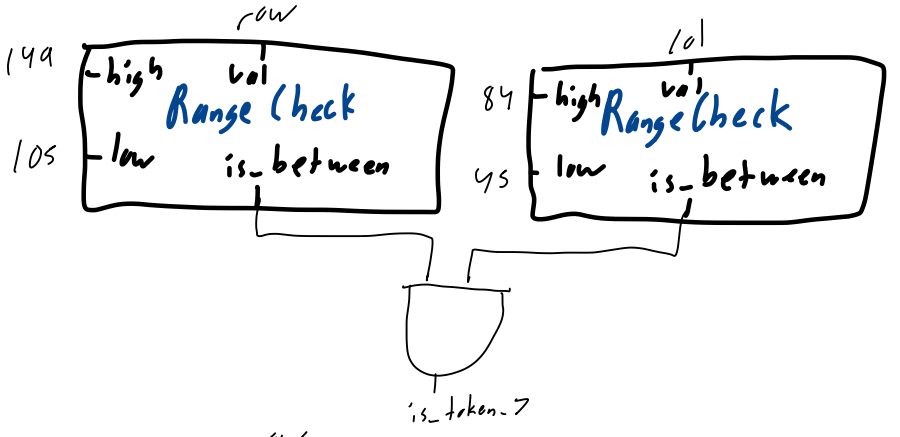

row
col
74
high
val
Range Check
30
low
is_between
339
300
is_token_3
424
385
is_token_4
507
470
is_token_5

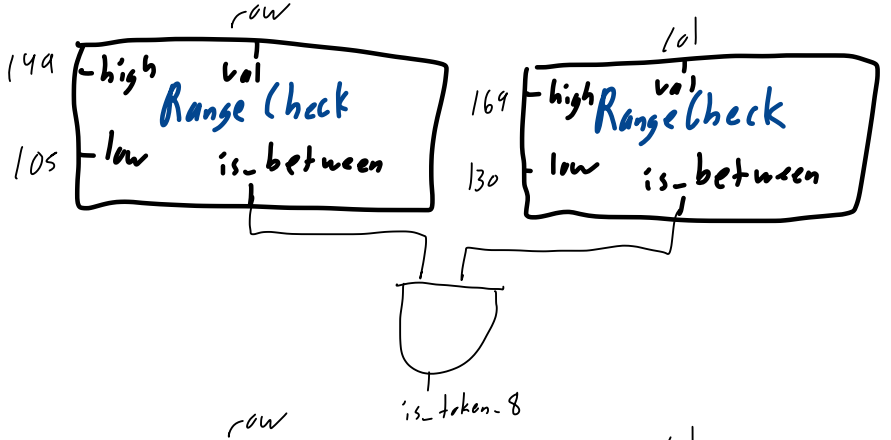
Token

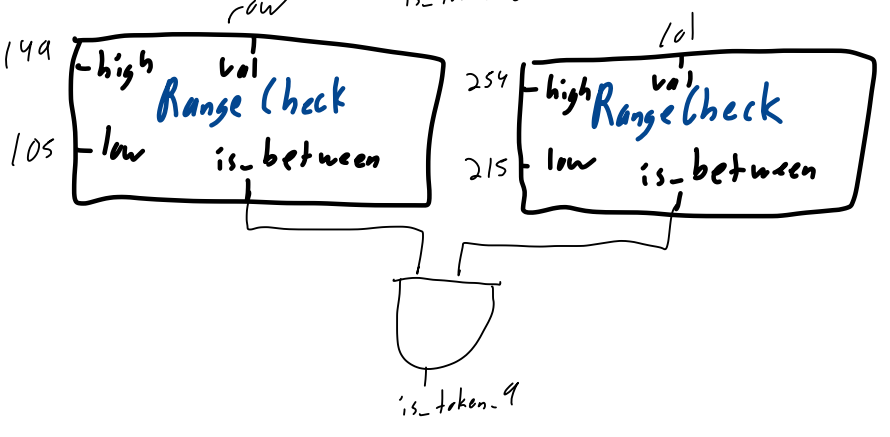


is_token_row_0 = is_token_0 | is_token_1 | is_token_2 | is_token_3 | is_token_4 | is_token_5 | is_token_6

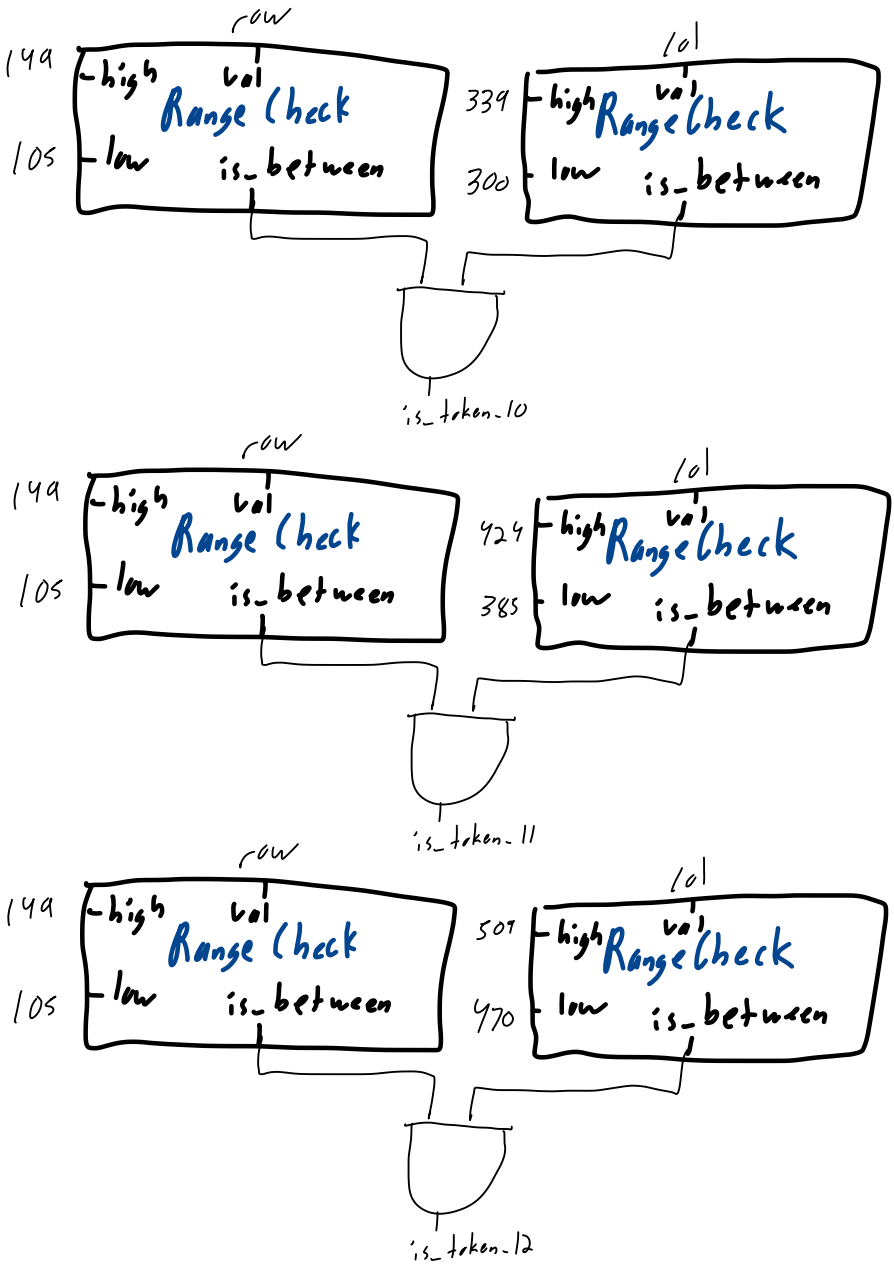
Token


raw
val
149
high
Range Check
105
low
is_between
101
val
84
high
Range Check
95
low
is_between
is_token_7

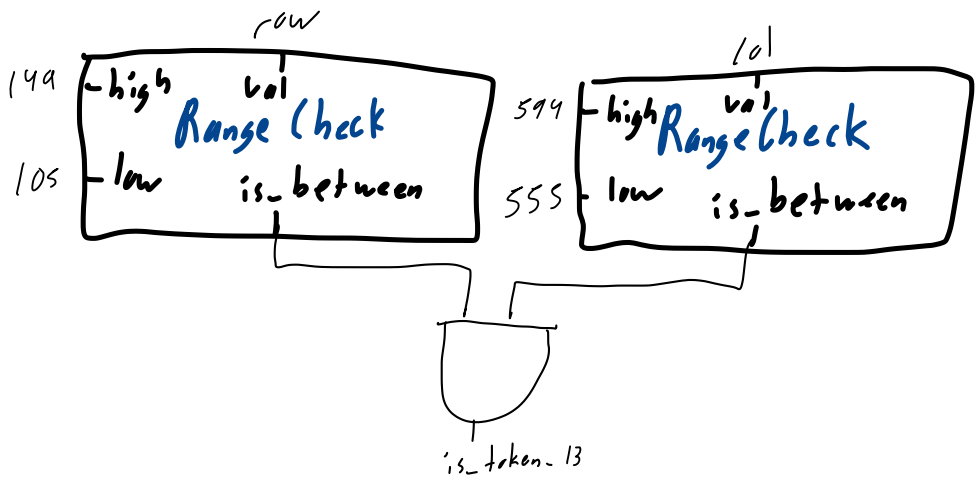

raw
val
149
high
Range Check
105
low
is_between
101
val
169
high
Range Check
130
low
is_between
is_token_8


raw
val
149
high
Range Check
105
low
is_between
101
val
254
high
Range Check
215
low
is_between
is_token_9

Token

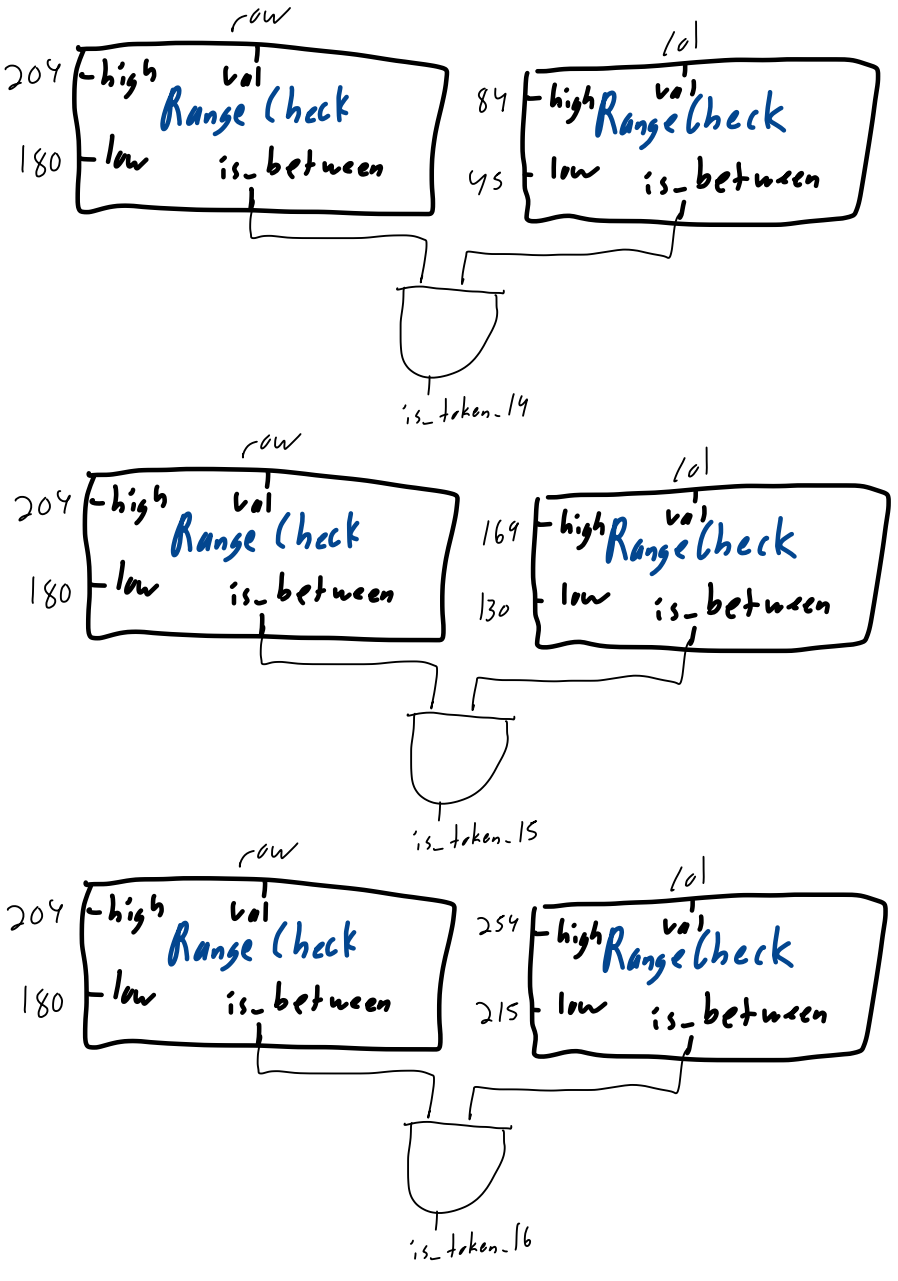

row
149
high
val
Range Check
105
low
is_between
101
339
high
val
Range Check
300
low
is_between
is_token_10
row
149
high
val
Range Check
105
low
is_between
101
424
high
val
Range Check
385
low
is_between
is_token_11
row
149
high
val
Range Check
105
low
is_between
101
507
high
val
Range Check
470
low
is_between
is_token_12

Token

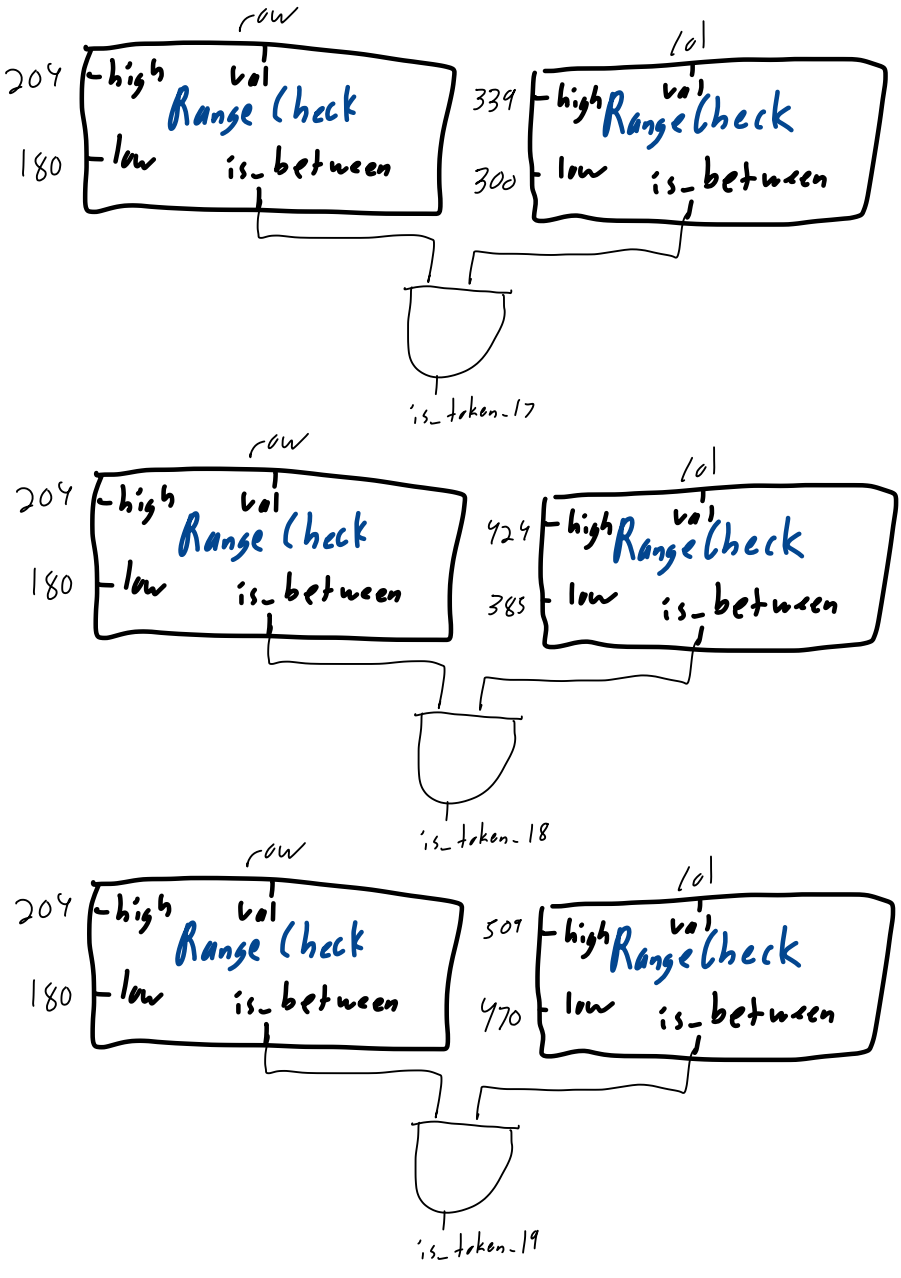


$$is\_token\_row\_1 = is\_token\_7 \mid is\_token\_8 \mid is\_token\_9 \mid is\_token\_10 \mid is\_token\_11 \mid is\_token\_12 \mid is\_token\_13$$

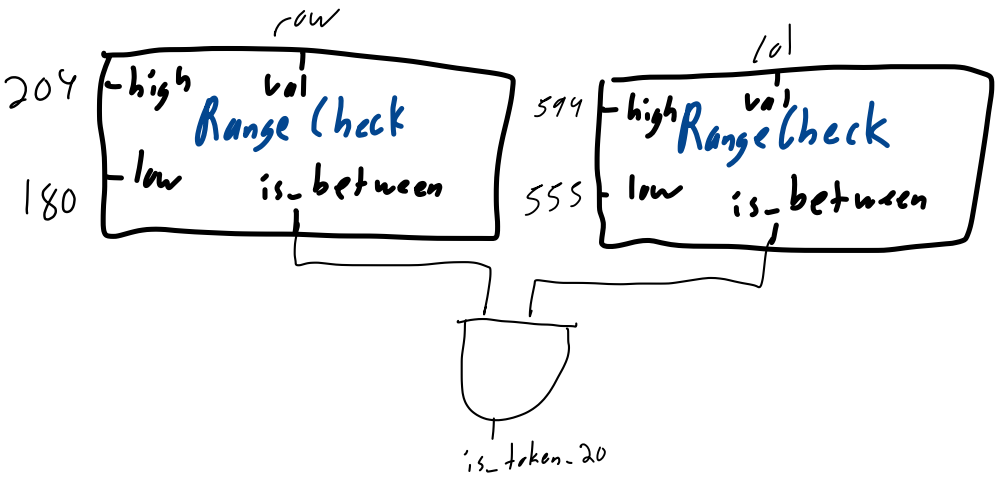
Token


raw
204
high
val
Range Check
180
low
is_between
col
84
high
val
Range Check
45
low
is_between
is_token_14
raw
204
high
val
Range Check
180
low
is_between
col
169
high
val
Range Check
130
low
is_between
is_token_15
raw
204
high
val
Range Check
180
low
is_between
col
254
high
val
Range Check
215
low
is_between
is_token_16

Token

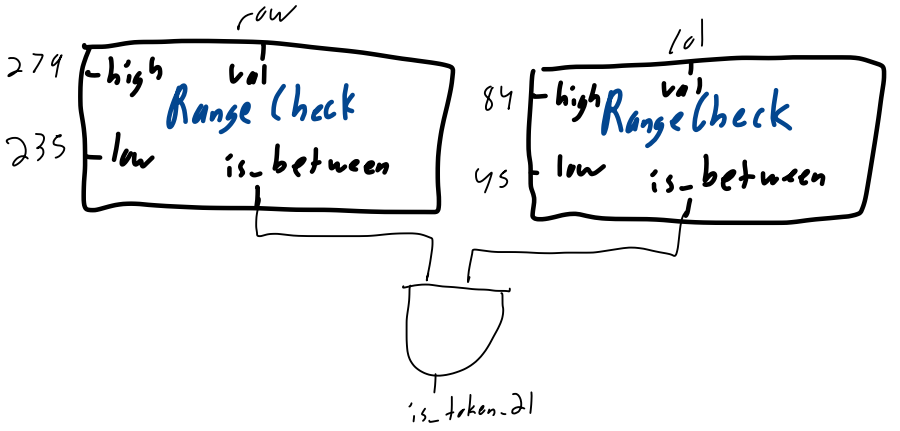

row
col
204
high
val
Range Check
180
low
is_between
339
high
val
Range Check
300
low
is_between
is_token_17
204
high
val
Range Check
180
low
is_between
424
high
val
Range Check
385
low
is_between
is_token_18
204
high
val
Range Check
180
low
is_between
507
high
val
Range Check
470
low
is_between
is_token_19

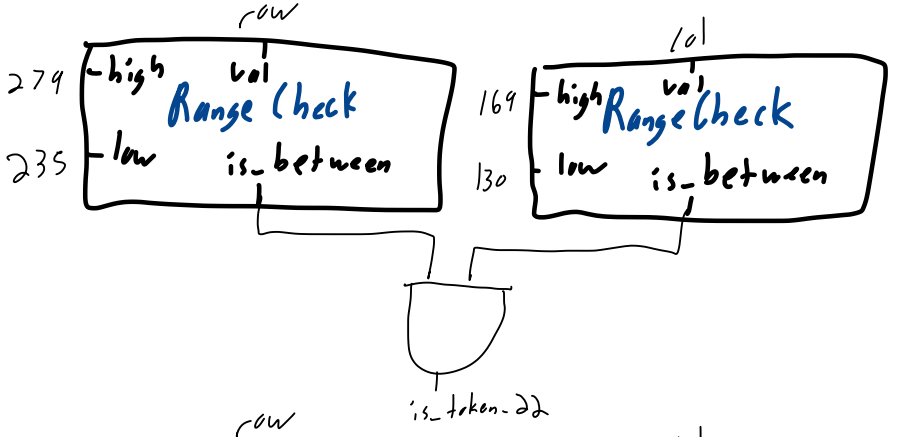
Tcktn

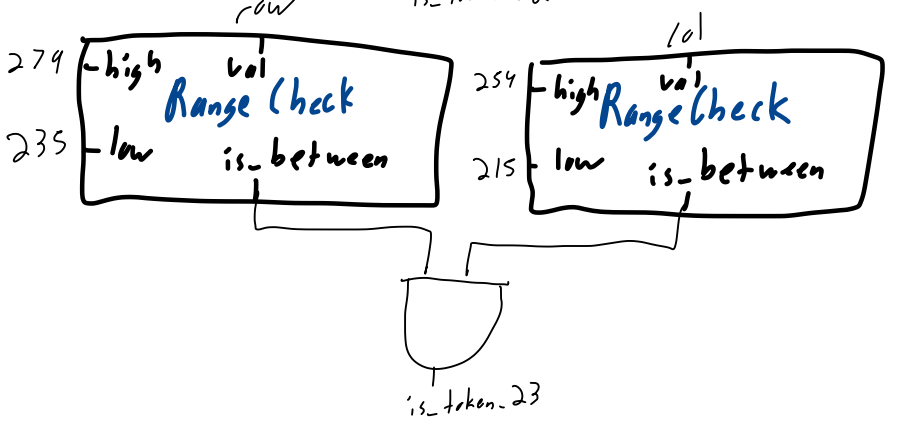


$$is\_token\_row\_2 = is\_token\_14 \mid is\_token\_15 \mid is\_token\_16 \mid is\_token\_17 \mid is\_token\_18 \mid is\_token\_19 \mid is\_token\_20$$

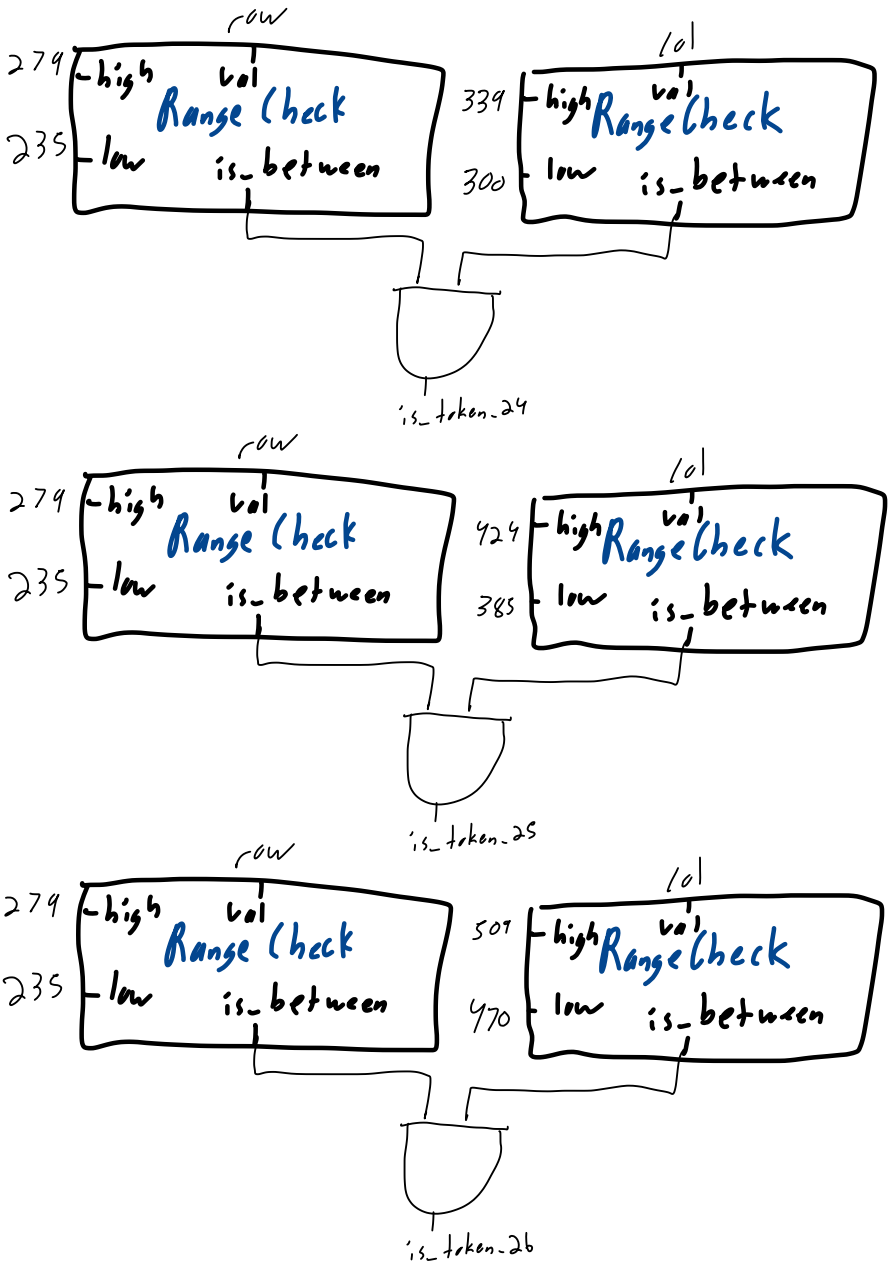
Token


row
col
279
235
high
low
val
is_between
Range Check
84
95
is_token_21

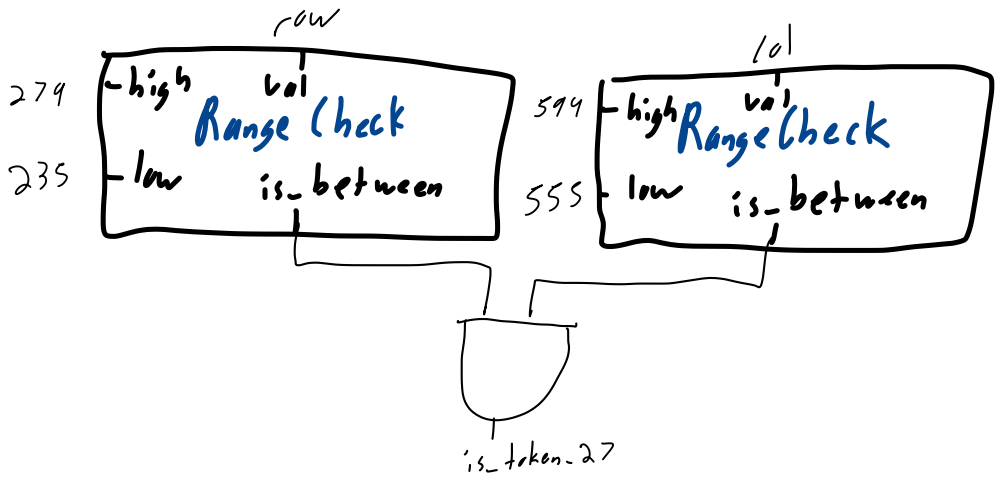

row
col
279
235
high
low
val
is_between
Range Check
169
130
is_token_22


row
col
279
235
high
low
val
is_between
Range Check
254
215
is_token_23

Token

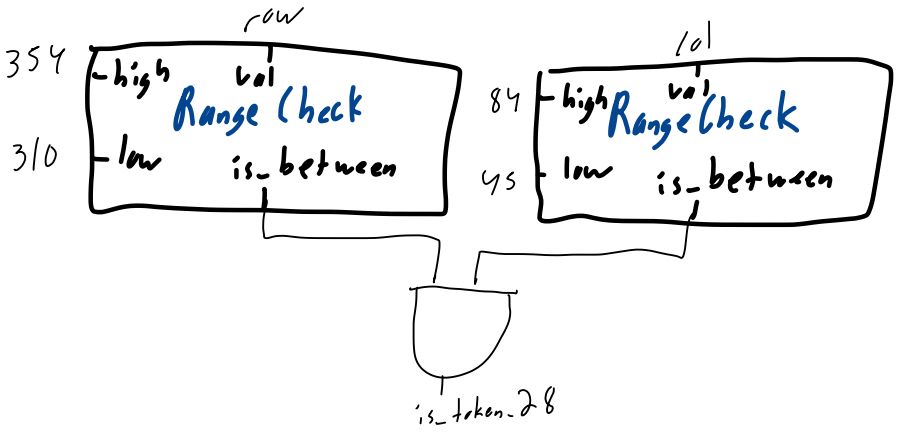

row
279
high
val
Range Check
235
low
is_between
col
339
high
val
Range Check
300
low
is_between
is_token_24
row
279
high
val
Range Check
235
low
is_between
col
424
high
val
Range Check
385
low
is_between
is_token_25
row
279
high
val
Range Check
235
low
is_between
col
507
high
val
Range Check
470
low
is_between
is_token_26

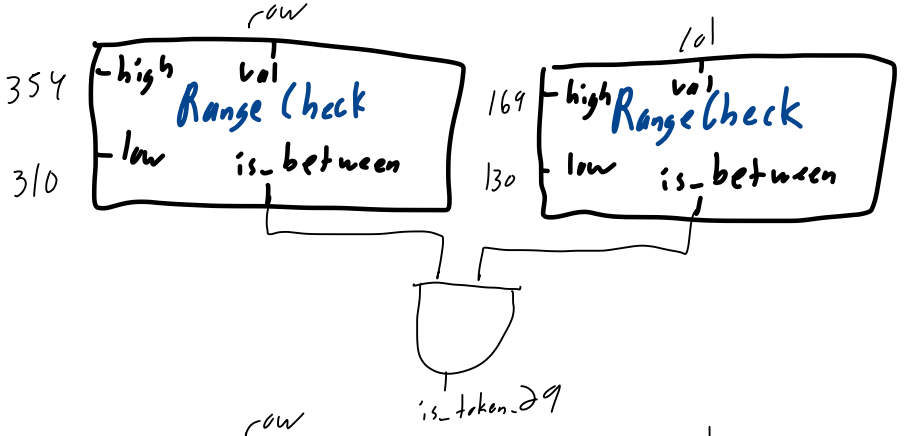
Token

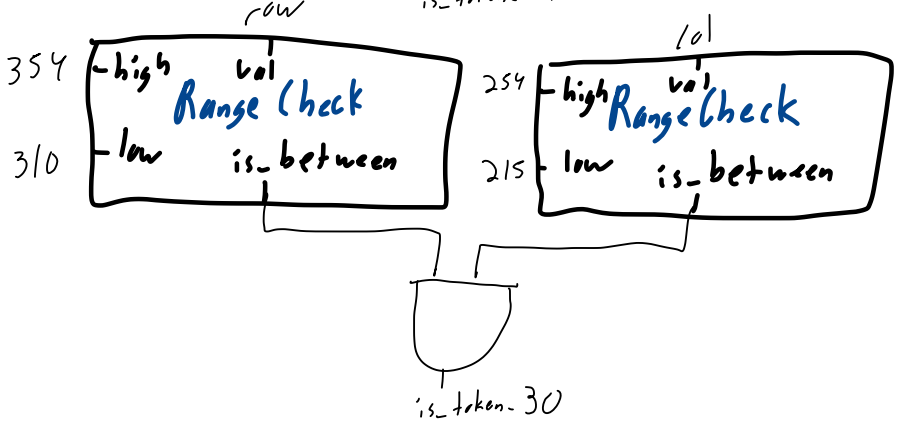


is_token_row_3 = is_token_21 | is_token_22 |
is_token_23 | is_token_24 |
is_token_25 | is_token_26 |
is_token_27

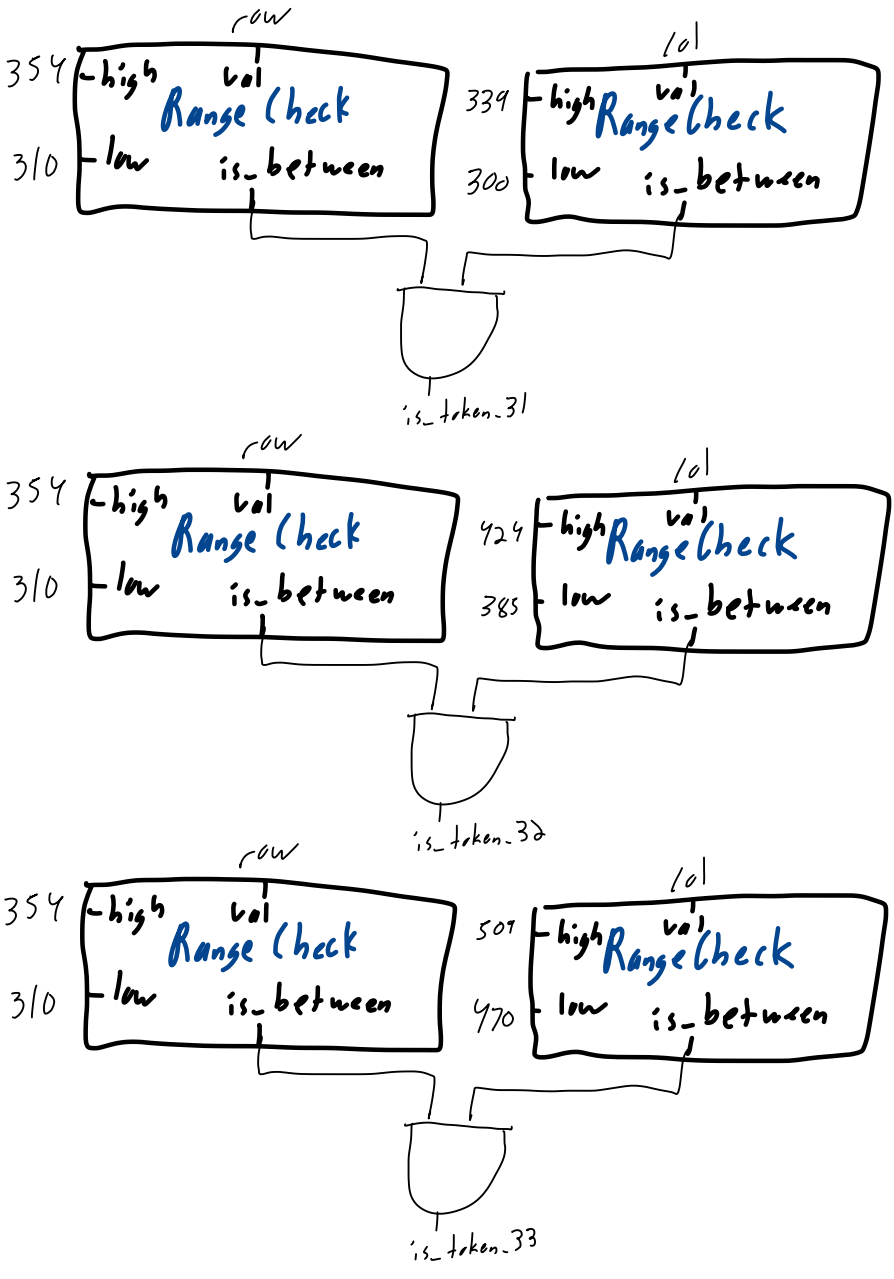
Token


raw
val
354
high
Range Check
310
low
is_between
lol
val
84
high
Range Check
45
low
is_between
is_token_28

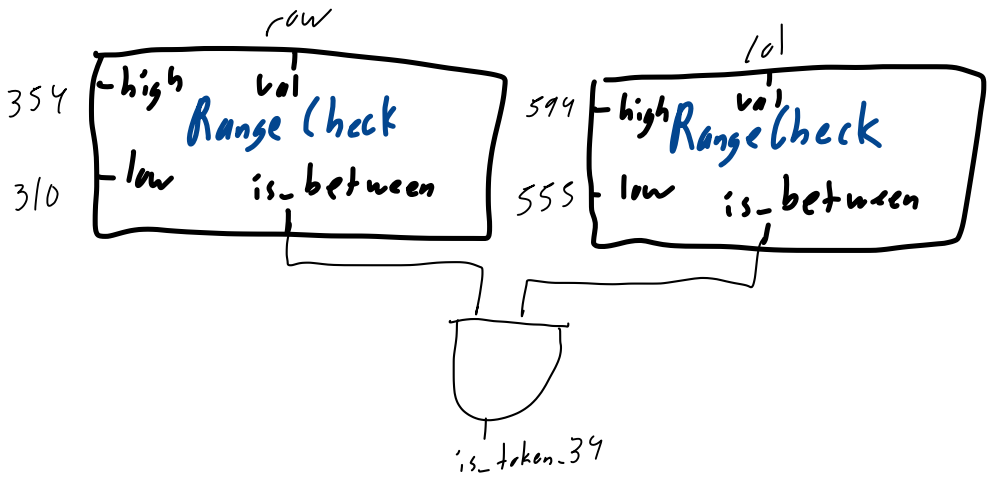

raw
val
354
high
Range Check
310
low
is_between
lol
val
169
high
Range Check
130
low
is_between
is_token_29


raw
val
354
high
Range Check
310
low
is_between
lol
val
254
high
Range Check
215
low
is_between
is_token_30

Token

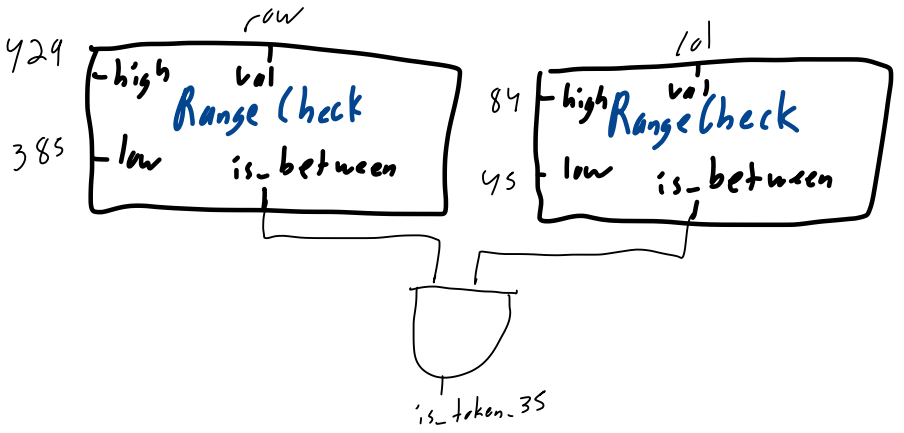

row
col
354
high
val
Range Check
310
low
is_between
339
high
val
Range Check
300
low
is_between
is_token_31
row
col
354
high
val
Range Check
310
low
is_between
424
high
val
Range Check
385
low
is_between
is_token_32
row
col
354
high
val
Range Check
310
low
is_between
507
high
val
Range Check
470
low
is_between
is_token_33

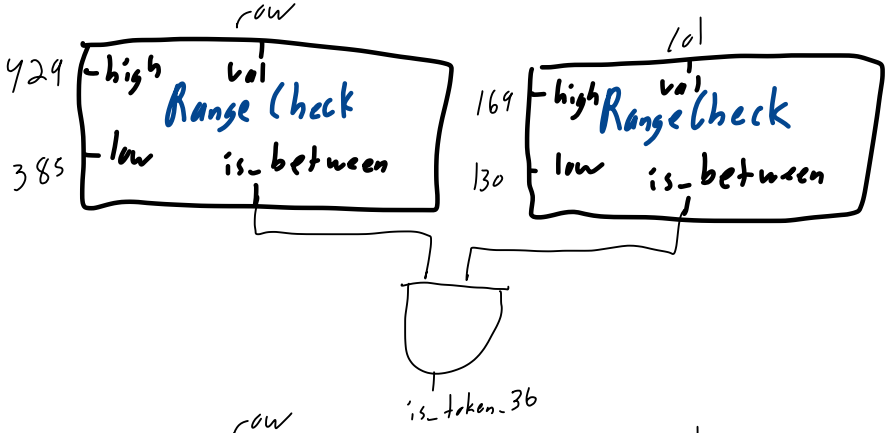
Token

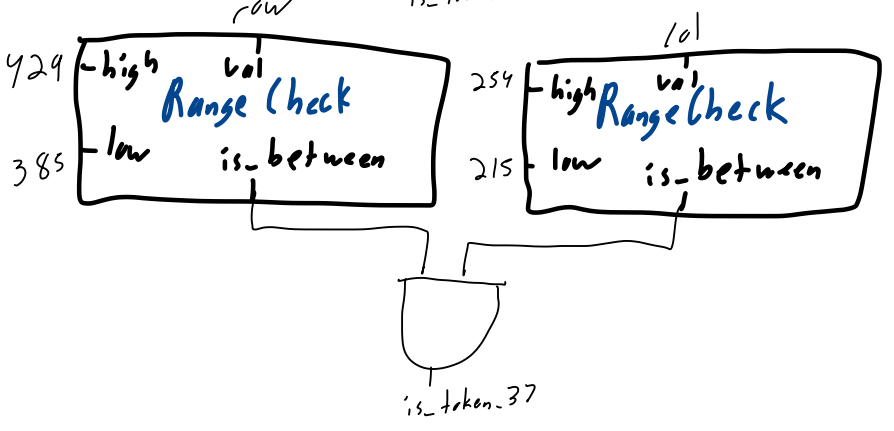


is_token_row_4 = is_token_28 | is_token_29 |
is_token_30 | is_token_31 |
is_token_32 | is_token_33 |
is_token_34

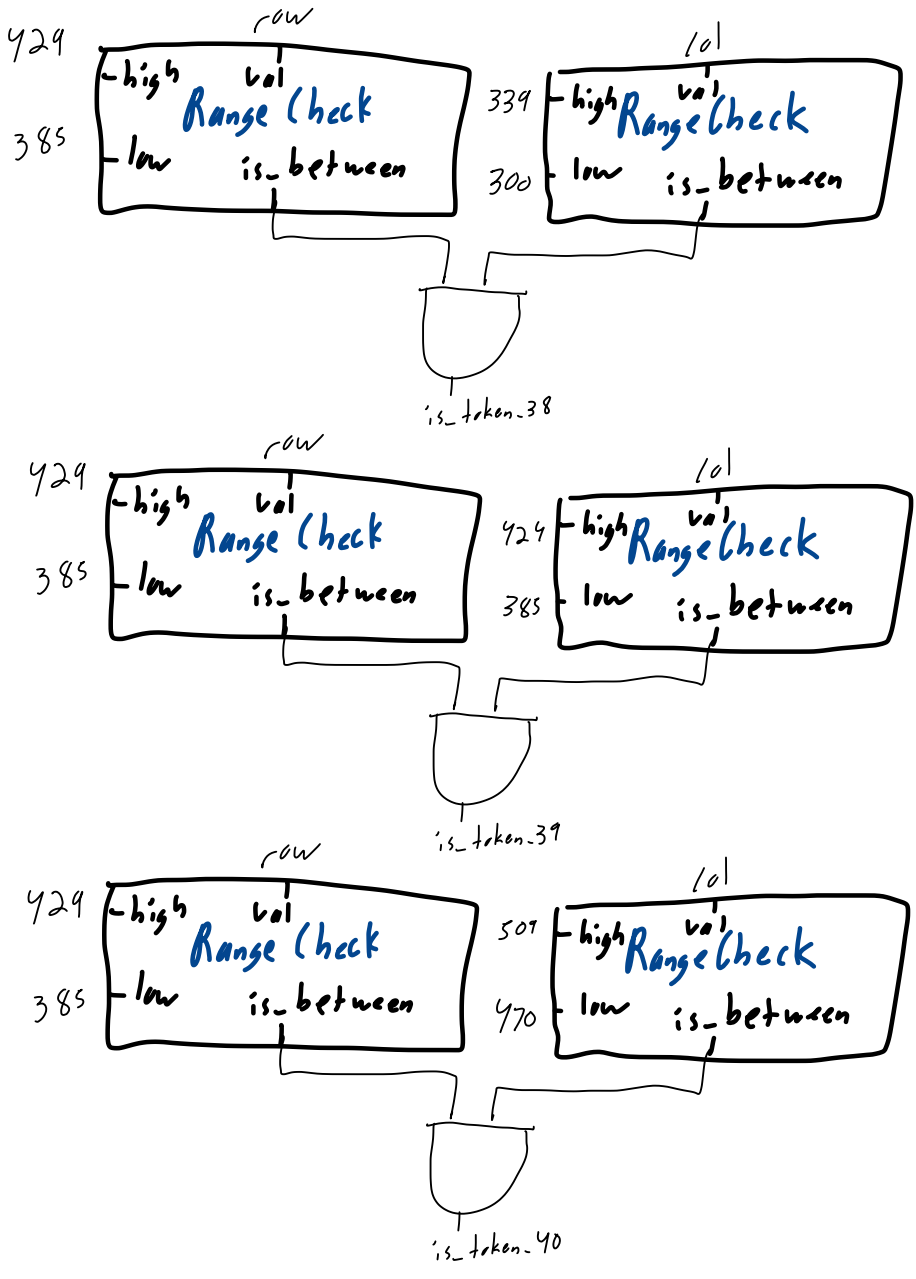
Token


raw
val
high
low
is_between
Range Check
429
385
84
95
is_token_35

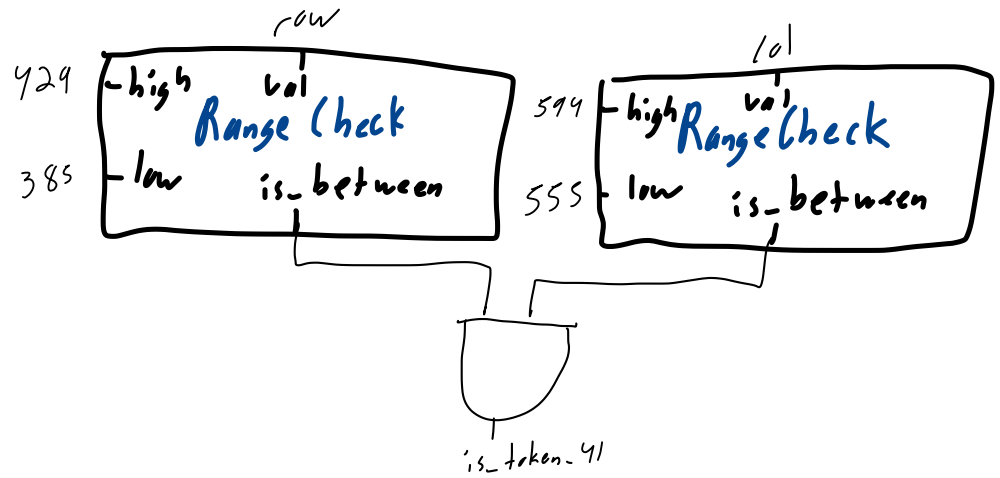

raw
val
high
low
is_between
Range Check
429
385
169
130
is_token_36


raw
val
high
low
is_between
Range Check
429
385
254
215
is_token_37

Token


row
429
high
val
385
low
is_between
Range Check
col
339
high
val
300
low
is_between
Range Check
is_token_38
row
429
high
val
385
low
is_between
Range Check
col
424
high
val
385
low
is_between
Range Check
is_token_39
row
429
high
val
385
low
is_between
Range Check
col
507
high
val
470
low
is_between
Range Check
is_token_40

Token



$$is\_token\_row\_5 = is\_token\_35 \mid is\_token\_36 \mid is\_token\_37 \mid is\_token\_38 \mid is\_token\_39 \mid is\_token\_40 \mid is\_token\_41$$

$$is\_token = is\_token\_row\_0 \mid is\_token\_row\_1 \mid is\_token\_row\_2 \mid is\_token\_row\_3 \mid is\_token\_row\_4 \mid is\_token\_row\_6$$

Ownership

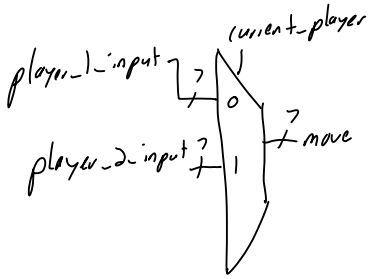
Determines Ownership of each token

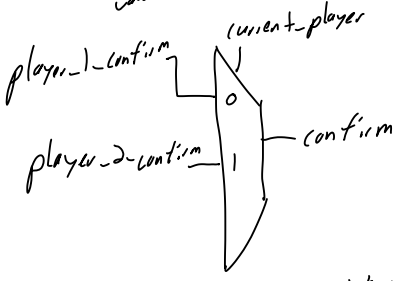
Inputs [6:0] player_1_input, player_1_confirm, [6:0] player_2_input, player_2_confirm

Outputs [1:0] tokens [5:0] [6:0]

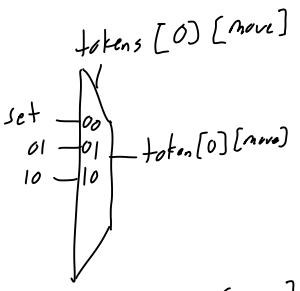
[1:0] tokens [5:0] [6:0] →

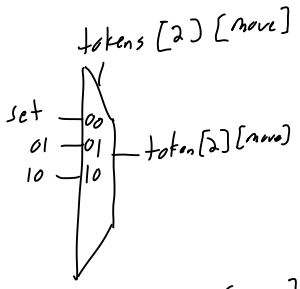
value 00 – unchosen
value 10 – player 2
value 01 – player 1
value 11 – not used

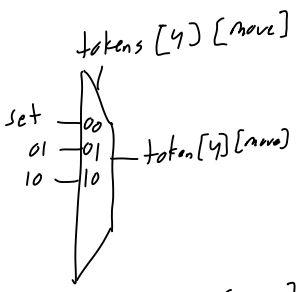


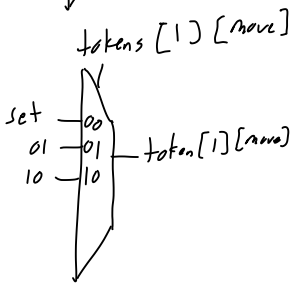


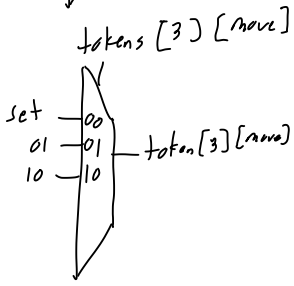
set = (~current_player) & confirm) ? 2'01 : ((current_player) & confirm) ? 2'10 : 2'00

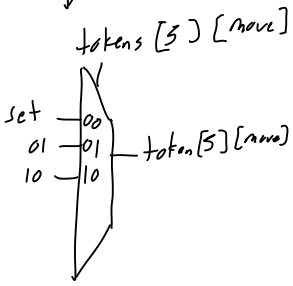






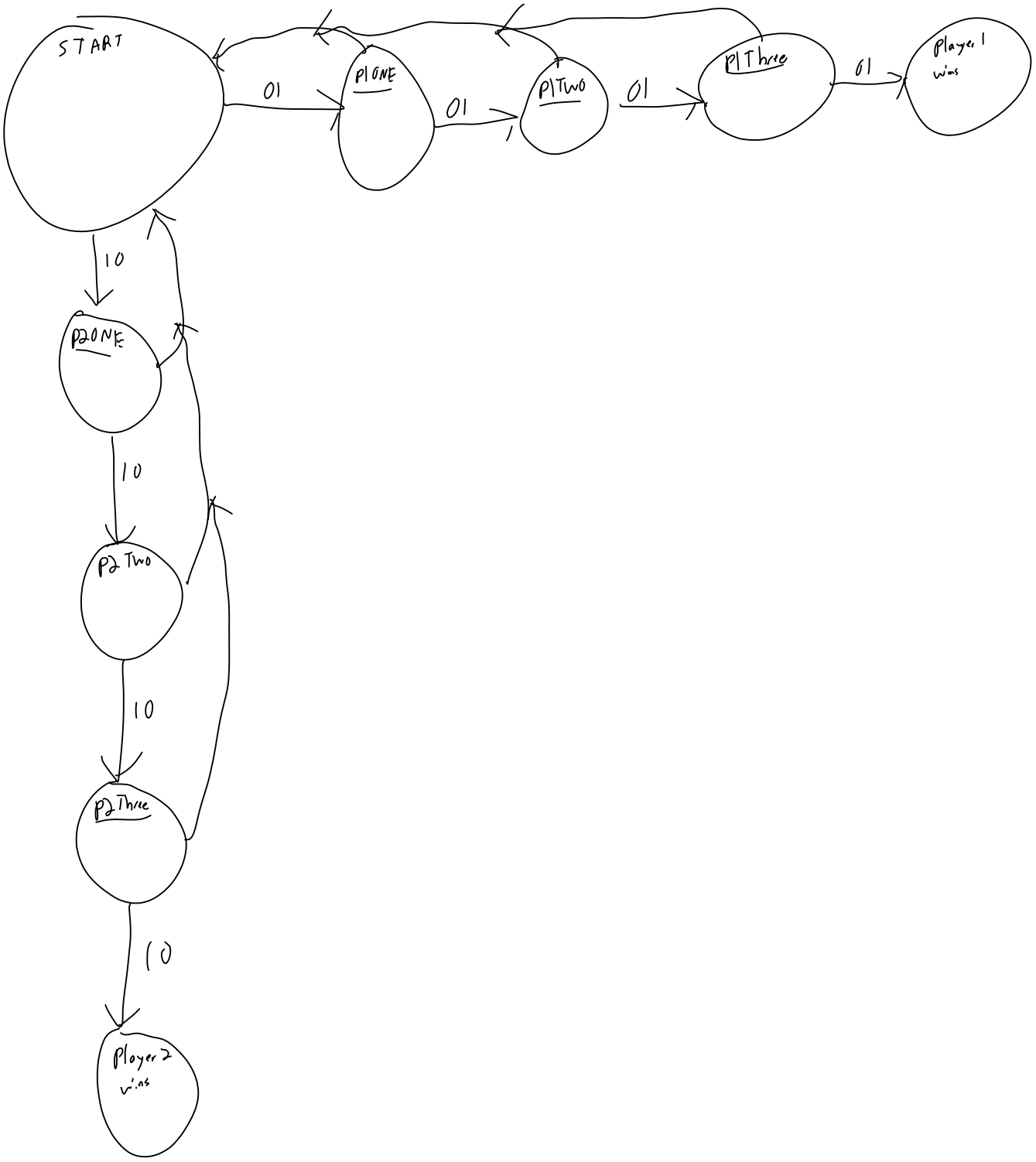




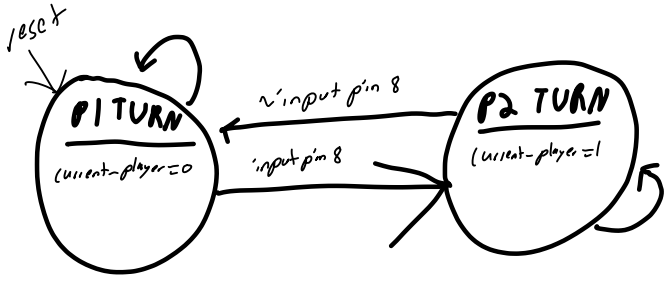


all these are chained so the logic is evaluated first (0, 1, 2, ...) and once set is assigned that is the end of the chained if else if ...

FSM win detector



FSM that determines which players turn it is



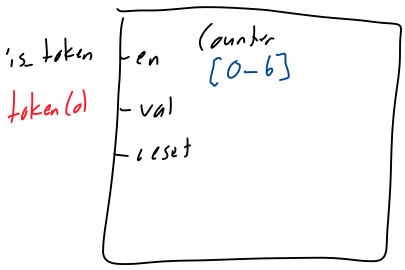
Color VGA_R, VGA_G, VGA_B

If is_board → = 00, 00, 11

If is_token & is_player_1 → = 11, 11, 00

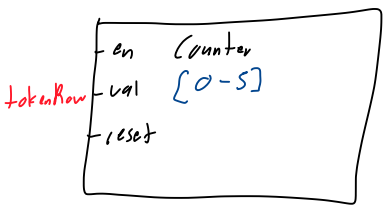
If is_token & is_player_2 → = 11, 00, 00

If is_token & (~is_player_1 & ~is_player_2) → 00, 00, 00



is_player_1 = (tokens[tokenRow][token(o)] == 2'01) ? 1 : 0

is_player_1 = (tokens[tokenRow][token(o)] == 2'10) ? 1 : 0

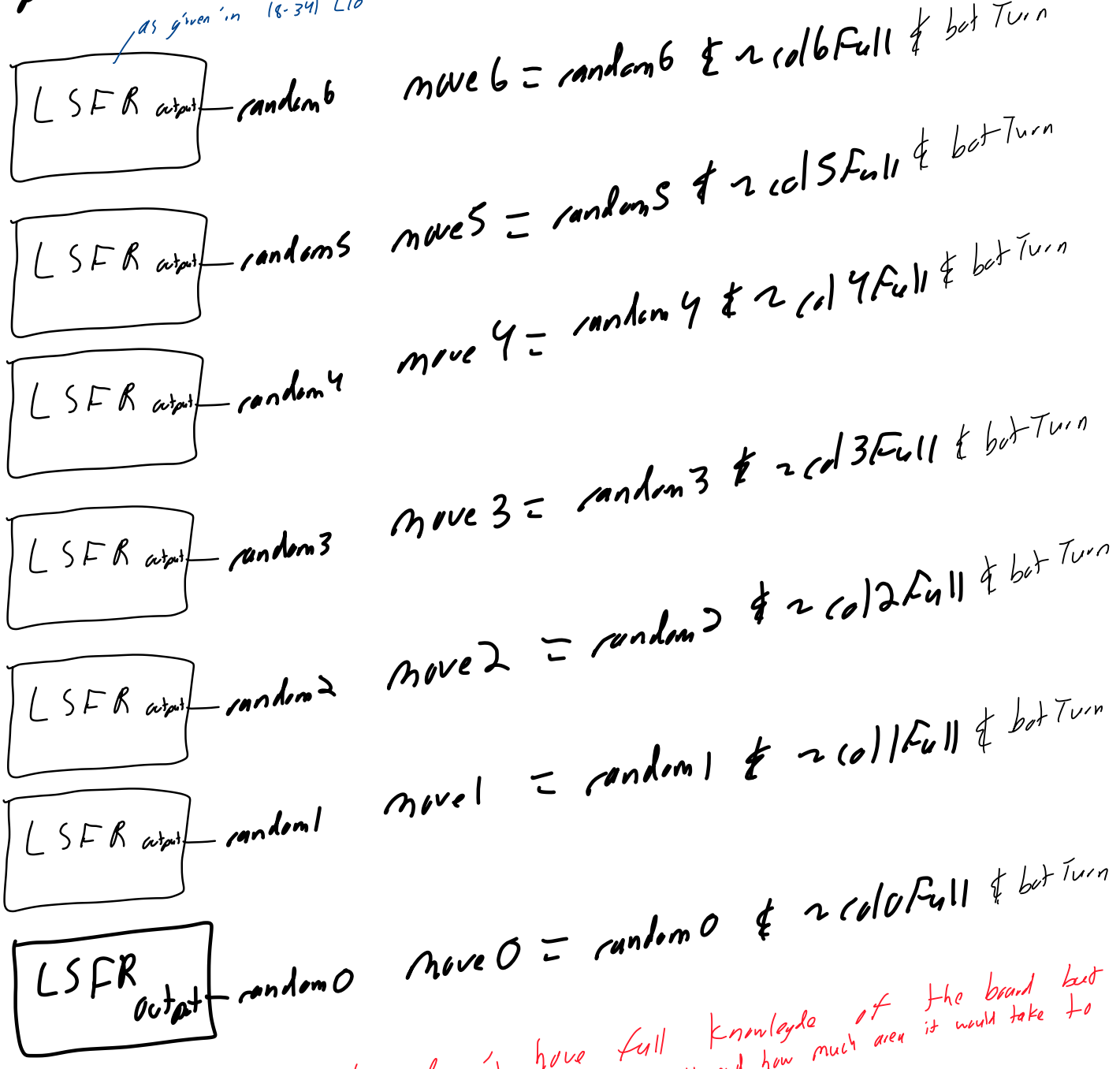


PvE

This is my idea for a very simplified PvE mode since after designing the other modules I am concerned about my remaining area.

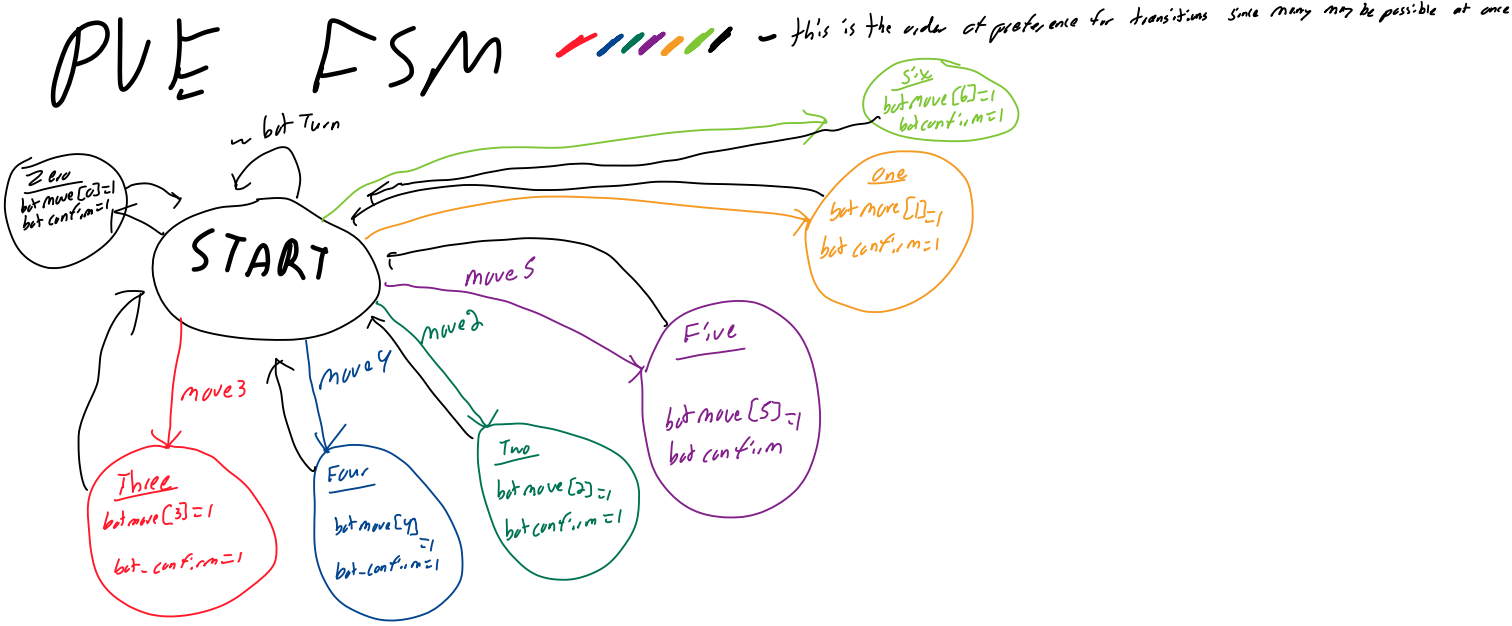
input col0Full, col1Full, col2Full, col3Full, col4Full, col5Full, col6Full, botTurn

outputs [6:0] bot_move, bot_confirm



One problem is that bot doesn't have full knowledge of the board bcz I am concerned about routing the 82 wires (42 tokens that need bits each) and how much area it would take to reason on them.

PVE FSM


- this is the order of preference for transitions since many may be possible at once
bot Turn
START
Zero
bot move [0] = 1
bot confirm = 1
Six
bot move [6] = 1
bot confirm = 1
One
bot move [1] = 1
bot confirm = 1
move 5
move 2
move 4
move 3
Five
bot move [5] = 1
bot confirm
Two
bot move [2] = 1
bot confirm = 1
Three
bot move [3] = 1
bot_confirm = 1
Four
bot move [4] = 1
bot_confirm = 1

Testing

VGA can be tested using concurrent assertions to ensure that its output waveform and row/col count is correct and then tested to see it actually outputs to a display.

Board will be tested via connecting to a display and checking that the board is shown correctly

Token will be tested by forcing is_player1 and is_player2 tol separately and checking that all tokens are correctly shown