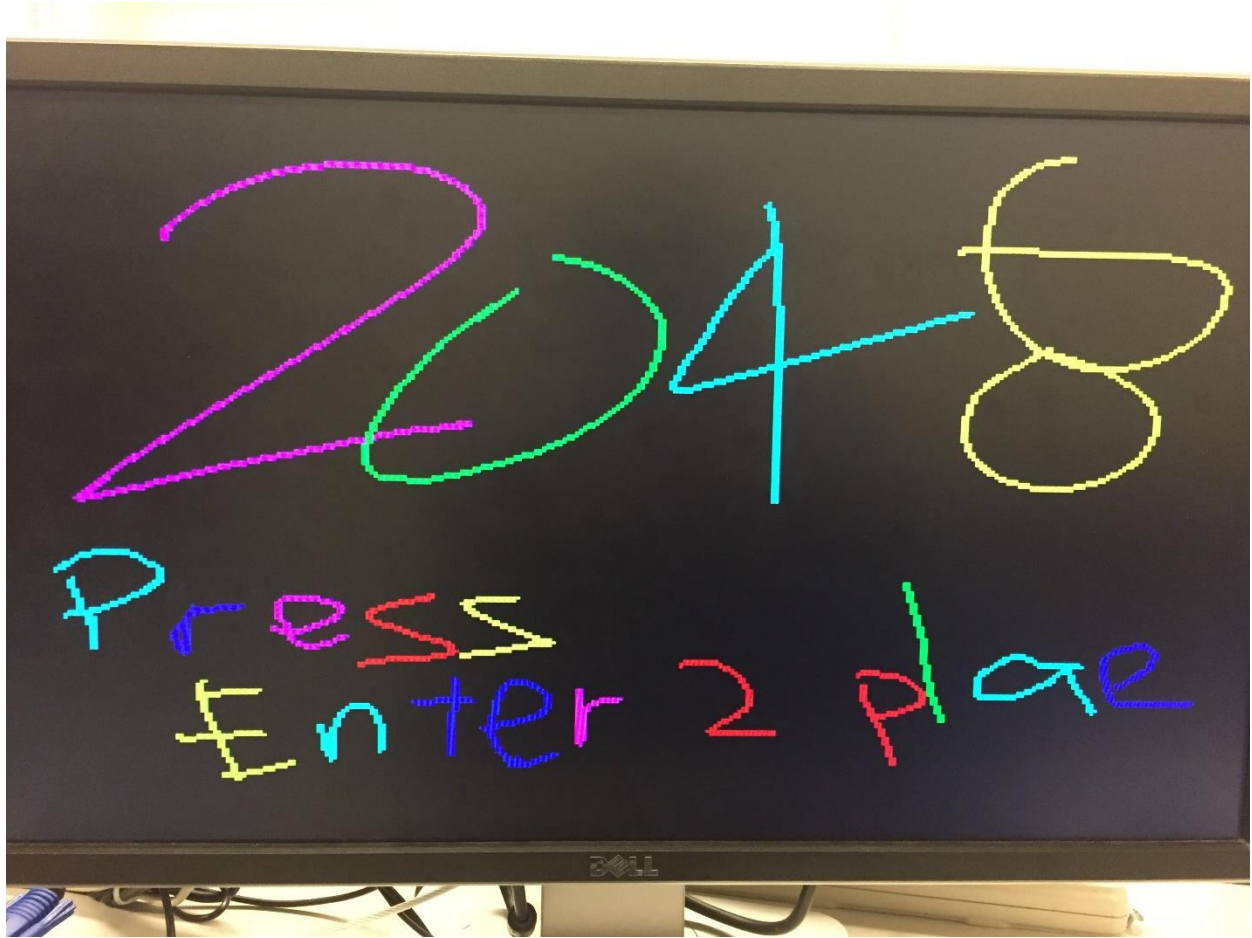


# ECE 241 H1 F



## Project – 2048 Game

Lab session

PRA 0104 Monday 3 – 6pm

TA

Ehsan Alimohammadian

Students

Sheng Zhao 1003273913

Yuwen Zheng 1001075463

# Introduction

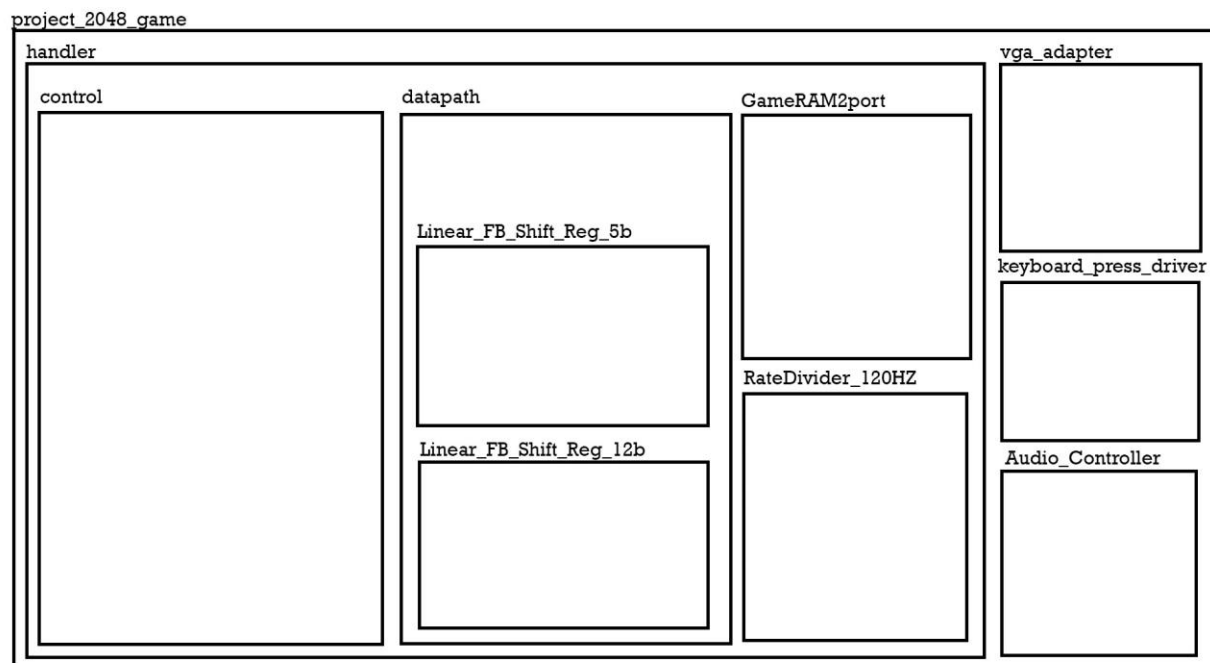
Our project was to implement the popular mobile game 2048 on the De1-SoC board. We have decided on this game due to its challenging game algorithm we needed to implement but it still being manageable as there are only 4 inputs from the user and only 1 winning and losing state.

The game is played on a 4 by 4 square board with the goal being to merge similar numbers, which are all powers of 2, to form the largest possible number. Each turn, the player can select up, down, left or right, which corresponds to a movement of all the numbers on the board in that specified direction. Any similar numbers who collide are merged to form a new number double the size (or sum), and a new random number is spawned on the board. The player wins if he attains the number 2048, and loses if there is no longer a valid move left on the board.

Many variants of the game exist, each having their own quirky mechanics, but for our project we will be looking at the original version [1].

In total, the project took about 60 hours to complete, a full timeline can be found in the appendix (Appendix A).

## The design



**Figure 1: Overview of Verilog modules**

The main modules of the game can be seen in Figure 1. Both VGA and audio controller modules are found on EECG's website [2], and the keyboard module is sourced from the University of Washington [3]. The handler module acts as a bridge for the 2 main

modules, control and datapath to communicate. Logic is handled within the control module, whereas all the actual computation is done in the datapath module. The rest are helper modules to facilitate computation. A more detailed explanation of all modules can be found in the appendix (Appendix B).

## Game Operation

The entire game's operation can be broken down into three parts to help understand how each part works and when they transition between the states.

### Game initiation

Upon programming the board, the FSM is initialized to the TITLE\_SCREEN state, which simply waits for a signal to start the main initialization sequence of the game. This coupled with the .MIF file used to initialize the VGA RAM forms our title screen.

Pressing the enter key (or SW[8]) starts the main initialization sequence. First the game RAM is cleared, done by having a counter looping through all 16 possible squares and writing each spot to have a value of 0, which signifies an empty spot. The address for the RAM is the coordinates of each square, with X begin the horizontal, Y the vertical, starting at 0000 from the top left to 1111 in the bottom right.

Next, the game will draw this empty board and the high score data to the screen. Simply put, a counter loops through then entire 320 by 240 pixels screen and at each pixel, the color determined by a (massive) set of if-else statements. For the text and borders, this is straight forwards and we simply check for coordinates corresponding to the pixels we need to draw (that is, a non-black color). For the actual game board, we would need to determine three things, which square is the current pixel in, what value is present in that square, and what is the effective coordinate of that pixel within the square. To find which square the pixel is in, with the knowledge that the board is 240 by 240 pixels large, and each square being 57 by 57 pixels (outer borders are 3 pixels thick and inner separation lines are 2 pixels thick), a simple if-else statement would give us the coordinates of the square. We then take this coordinate and retrieve the value present in the square. Finally, an equation is used to find the effective X and Y coordinates within said square (i.e. a X coordinate of 180 on the screen would be effectively the X coordinate of 0 in the third column of squares). With all this information we can finally traverse the massive if-else statements to decide on the color to plot. This method is rather complicated and requires many more states than if ROMs are used, which will be further discussed in a later section.

After drawing the empty board, a new number must be spawned in. The coordinates are the 4 most significant bits of a pseudorandom Linear Feedback Shift Register of 5 bits. 5-bit LFSR is required since having 4 bits would mean that coordinate 1111 can never be chosen since LFSR cannot generate an all 1s output (which would also stop it from working further). Every time a new number is chosen, the corresponding spot on the

board is checked, if it is empty, we can spawn in the number and carry on, else we get a new one.

Finally, we draw in the random number. This draw process is separated for it to be animated. A counter (cascading counter) counts from 0 to 57 and a similar counter (initialized to the square of this random number) counts through the pixels. However, each cycle we only draw up to the number in the cascading counter, after which we increment the cascading counter and wait for a 120 Hz signal for the next cycle. As a result, we can animate in the newly spawned number.

## **Game algorithm**

After the initialization process, we end up in the `GAME_WAIT_FOR_MOVE` state, where we wait for a user input. Upon receiving a user input (and the release of the signal), we begin the calculation process. Due to symmetry of the 4 different moves, we will just explain one of the possible moves, UP.

To process the move UP, we will be checking the board towards the opposite direction, that is, top to bottom. This will ensure that we always move the numbers first, instead of merging them in the middle of the board, which may lead to wrong calculations. Note also that since we are moving in the vertical orientation, the order at which each square in each row is checked is irrelevant, but for convenience sake we will always be doing the non-important axis (the one we are not moving in) in an incremental fashion (from 00 to 11).

Now for the logic, for each square, there can only be 3 possible next steps, either the number does not move, the number moves and merges with the next square, or, the number simply moves to the next square.

For a number which does not move, it could be either that the number is in a square along the boundary (in our case now, the top row), or the square is next to another (in the direction the user specified, i.e. UP) containing a different number than the current one.

For move and merge, the next square must be of the same number.

For simply move, the next square must be empty.

To implement this logic, we set up two counters. An iteration counter for the number of times the board has been checked and another for looping through all the squares. Every square we check for the 3 types of next steps mentioned above and the respective signals are switched on to select our next state. For no move, we simply move to the next square. For merge moves, we increment the next square (a single left bit shift since our numbers are all powers of 2) and then update the current to 0. For just move, we just update the next with the current value and set the current square to 0.

This process is required to be repeated for a maximum of 4 times before all the possible updates are made. A few more were added for insurance and to allow the high score counter to update which each user input (instead of the next). High score is simply stored in a register and updated as we retrieve each square's value.

After this, the same process is repeated for drawing, spawning a new number, and drawing that again in an animated fashion.

## **Game end**

Failure state detection for the game is slightly tricky, to lose the user must have no more valid moves in any direction. This has proved to be slightly too complicated and time consuming to implement fully (coupled with other features and debugging). Therefore, a slightly less vigorous method is used.

Recall that the random number is spawned into a square whose address is generated by a 5-bit LFSR. A LFSR is pseudorandom and repeats its number sequence every  $2^n - 1$  numbers (for a n-bit LFSR). As a result, if after 31 cycles (less since we only have 16 squares) and still we could not spawn a new number, the board must be filled, and we called this our game losing condition. This condition may still be triggered even though there are moves left, and is therefore not a perfect check (relatively good enough assuming the player does not make a bad move). The current implementation checks for 100 cycles before signaling for the losing screen to be drawn.

The final states of the game draw the losing screen in the same way as the main game and awaits a signal from the enter key to reset the game.

## **Report on success**

### **What worked**

The project has been a huge success. A working game was produced that worked as intended and the same as the original (with the quirk of numbers being able to merge more than once per move). We successfully implemented title screens, keyboard inputs, VGA display, audio (a beep tone unique to each input), the game logic with a hardware design language, high scores and a losing screen. The game flows naturally from starting the game with enter, to playing with the arrow keys, to losing and resetting with another tap of the enter key. All photos can be found in the appendix (Appendix C)

### **What didn't work**

Throughout the entire development process there had been numerous bugs that were rather difficult to find out. Majority were fixed after spending hours looking through ModelSim simulations. However, there was a couple that have yet to be resolved.

During the testing of the game, the game RAM would sometimes be randomly wiped clean upon a move that cause numbers to merge. To find the cause of the problem would prove to be rather difficult due to the random nature of how new numbers are

spawned in. The problem does not always present itself with the same set of inputs. Removing the random number portion cause the problem to go away entirely. The only way to pin point the problem would require a video capture of a game play that had the problem, then figure out a way to have the numbers spawn in the same order, and finally run it through ModelSim to be debugged. Another similar issue was that numbers occasionally “teleports” to another square upon merging in the top left square. Both issues seem to come and go with each compilation of the code, and did not happen frequently enough to be worth the time to investigate.

One possible explanation for the game RAM clear could be that since the default next state of the FSM was set to initialization, the game may have gotten itself stuck or jumped out of sequence of the states. This then led to the state to be set back to the default, causing a reset of game RAM.

## **What would you do differently**

We started the project with only a basic overview of how the entire game would operate. As a result, improvisation and changing things on the fly were a continuing theme throughout the coding of the game.

One of the main things was not using ROMs for display. This meant we had to plot each pixel that would be drawn on the screen, with the exception of the title screen. We made the decision to do so at the time (at when we thought was nearing the end of the project, which in hindsight was still early) because the structure of the states was not set up for the usage of ROMs. Doing so would require a time-consuming operation to restructure the flow of the state logic, which now in hindsight would have taken much less time compared to writing literally thousands of if-else statements.

Yet another issue was due to insufficient planning. RAM was used to store the gameboard, however, we had overlooked the simple fact the it would take another clock edge after we accessed it with an address. A decent amount of debug time could have been avoided had we simply planned better and ensured that we understood each component of the game fully before coding.

## **Conclusion**

This project component of the course has been beneficial in the overall understanding of how hardware design would look like. It serves both as a revision of all the other labs, and as a new learning experience of how each small component we had built before can work and communicate with one another.

## References

- [1] <https://gabrielecirulli.github.io/2048/>
- [2] [http://www.eecg.toronto.edu/~pc/courses/241/DE1\\_SoC\\_cores/](http://www.eecg.toronto.edu/~pc/courses/241/DE1_SoC_cores/)
- [3] <https://class.ee.washington.edu/271/hauck2/de1/index.html>

## Appendix A: Rough timeline of project development

31 Oct

- finalized on 2048 + entered into system

6 Nov

- submitted lab 7 + set up meeting on 7 Nov
- completed planning night of 6 Nov
- preliminary FSM, modules structure layout
- game logic worked out

10 Nov

- coding began evening

11 Nov

- lab partner came over, finished draft 1, partial working game logic with partial display working (dots for debugging)

Week of 13 Nov

### Issues

- game works but did not account for RAM I/O clock timing -> buggy logic
- did not have state to wait for key release -> hundreds of steps per user move input
- display math not working (due to RAM issue) -> display offset wrong
- infinite loop due to RNG not covering all spaces (which was further due to how the FSM was set up, same few numbers were read each time)
- 4bit LFSR does not allow for 1111

### Solutions

- added multiple signal extension states for extra time to resolve RAM I/O issue
- 1 state to wait for release of key
- updated the math equations used to generate effective X and Y coordinates
- now LFSR only enabled each time a new number is needed
- changed to 5 bit LFSR, 4 most sig bits are used

### New implementations

- VGA numbers added (non ROM)
- Keyboard input added



Week of 20 Nov

#### Issues

- game ram auto clear when sending to the right, occurs when the right columns are full and merging occurs, possibly due to fast clicking?
- enter does not send to clear board properly if current state initialized to title screen
- cascading animation for newly spawned numbers not working

#### Solution

- issue occurs extremely rarely, unable to replicate for debugging, unresolved
- was no longer an issue after fixing other parts
- number iterations and if-else statements were buggy, resolved animation issues

#### New implementations

- sound added
- randomized color for everything on screen upon each new move, toggle-able
- auto-play feature (failed)
- game lose detection
- new title screen, losing screen, high scores

## Appendix B

This section will be describing each module that is present and their purpose. For the ease of understanding, we would be separating the entire project into two big sections, the core and peripherals, described in their relevant subsections below.

### Core modules

The core modules are the ones responsible for handling user input, game logic and output of display.

#### VGA adapter module

This is the provided VGA adaptor module from Lab 7 used to drive the display.

#### Handler module

This module acts as the bridge for communications between the control and datapath modules. Helper modules such as rate dividers and the game RAM are also instantiated at this level.

#### Rate divider module

This module generates a 120 HZ clock signal to help operate the FSM to animate the newly generated number.

#### Game RAM

This 12-bit wide, 16 word RAM contains the current state of the game. Address for each square of the board is used to access the relevant number on it, which is stored in each word in the RAM. This is a 2 port RAM, one used for the actual updating of the game, the other is connected to switches and hex displays for debugging purposes.

#### Control module

This module contains the Finite State Machine and therefore all the logic required to operate the game logic and VGA display. States are transitioned based on user input and signals from datapath module.

#### Datapath module

This module does the actual computation, updating, data input and output of the game RAM.

### Peripheral modules

The peripheral modules are ones that adds value to the project but are not critical for the operation of the game.

**Hex display modules**

These modules are used to display the selected data from the game RAM debugging purposes.

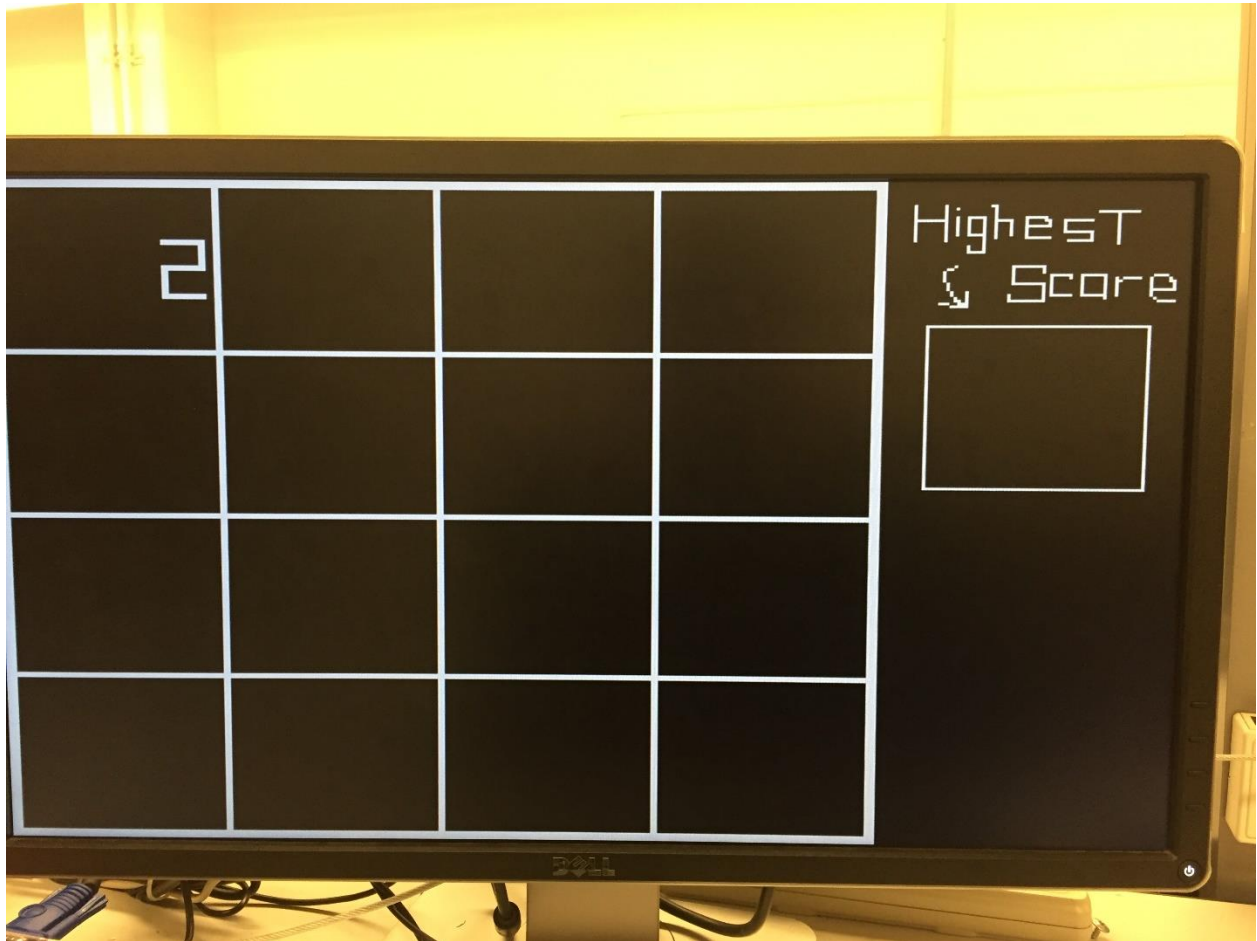
**Keyboard module**

This module decodes the inputs of PS/2 keyboard which are used to control the game [3].

**Audio module**

This is the provided Audio module on piazza used to output sound when the user keys in an input.

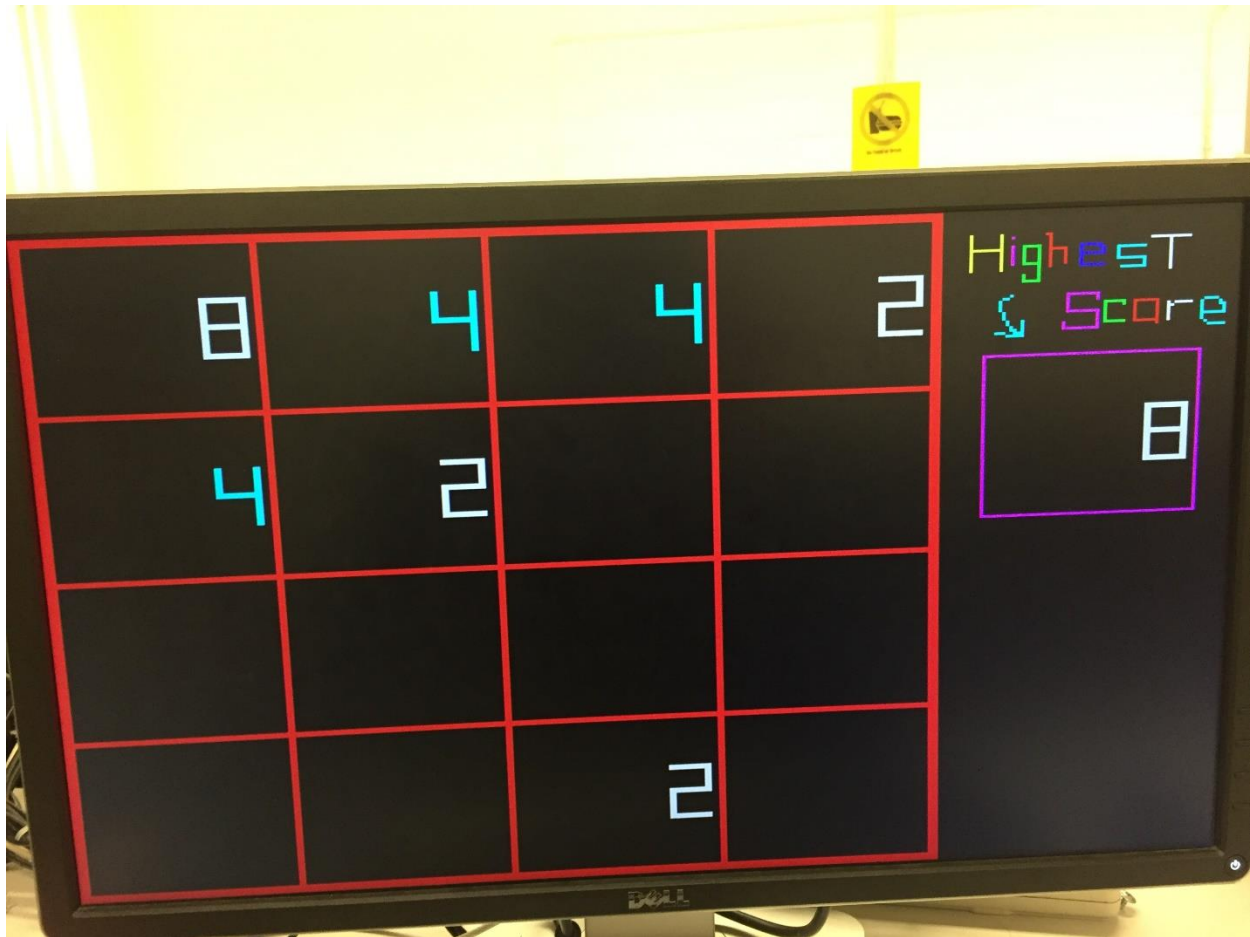
## Appendix C: Game photos



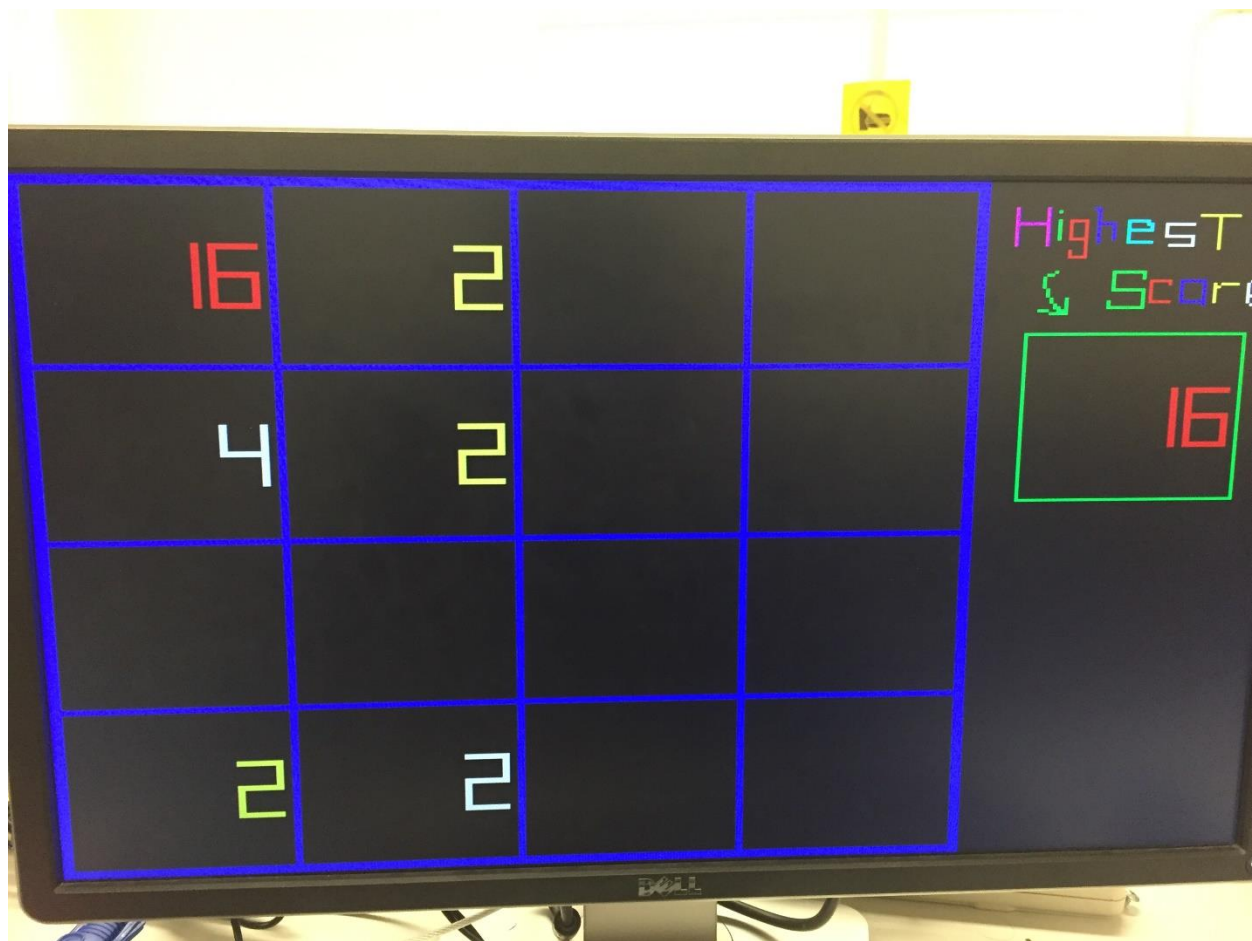
a. New game, waiting for user input



b. HEX displays are made to show game RAM's data via the second port



c. Game has randomized colors

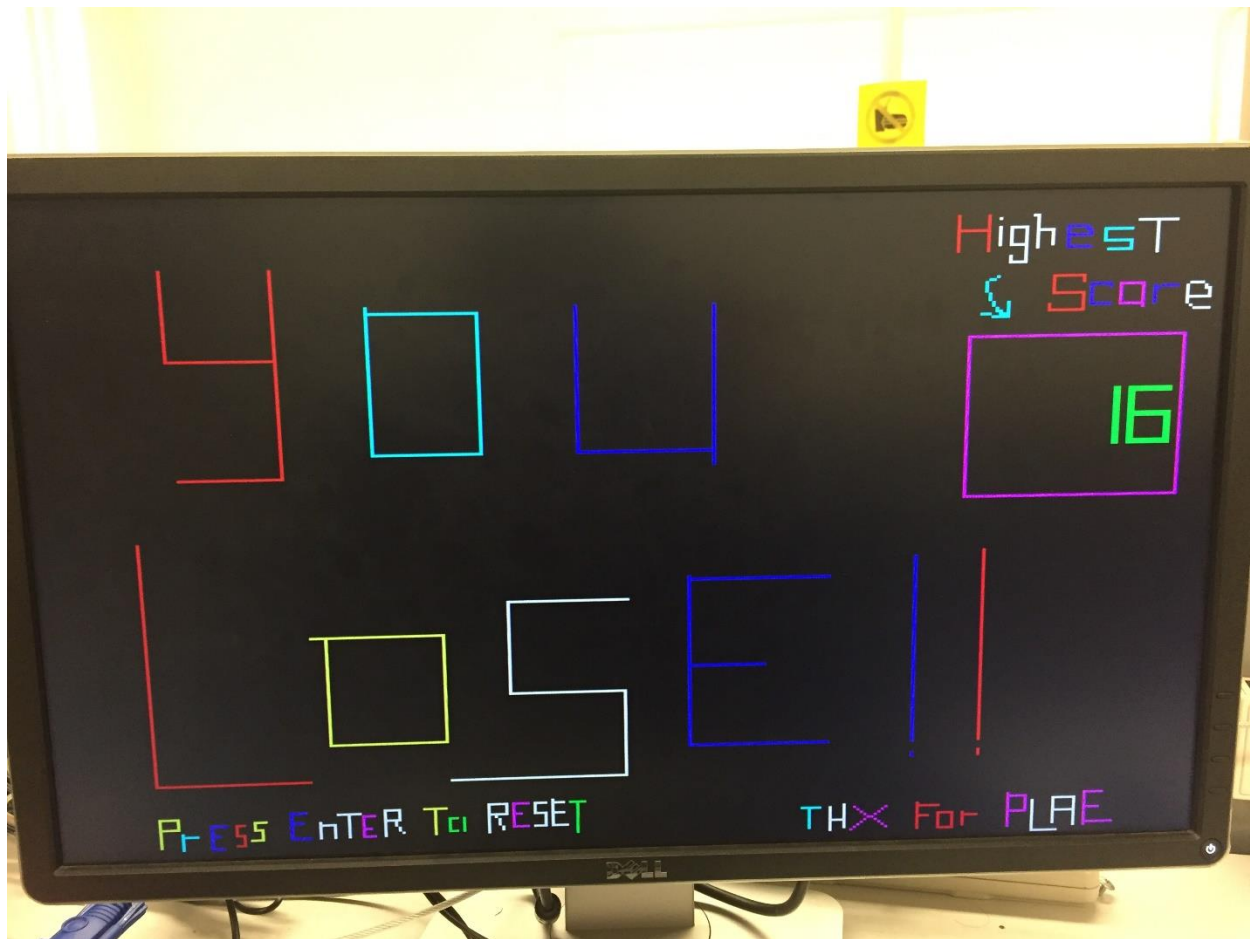


d. Newly randomized colors for each move



e. No more possible moves for UP, if pressed, game treats it as a loss





f. Losing screen, each letter is a different color each time, enter key allows quick reset

## Appendix D: Verilog Code

// Part 2 skeleton

```
module project_2048_game
(
    CLOCK_50, // On Board 50 MHz
    // Your inputs and outputs here
    // The ports below are for the VGA output. Do not change.
    VGA_CLK, // VGA Clock
    VGA_HS, // VGA
H_SYNC
    VGA_VS, // VGA
V_SYNC
    VGA_BLANK_N, // VGA BLANK
    VGA_SYNC_N, // VGA SYNC
    VGA_R, // VGA Red[9:0]
    VGA_G, // VGA
Green[9:0]
    VGA_B, // VGA
Blue[9:0]
    KEY,
    SW,
    LEDR,
    HEX0,
    HEX1,
    HEX2,
    PS2_DAT, // PS2 data line
    PS2_CLK, // PS2 clock line

```

```

        AUD_ADCDAT,

        // Bidirectionals
        AUD_BCLK,
        AUD_ADCLRCK,
        AUD_DACLK,

        FPGA_I2C_SDAT,

        // Outputs
        AUD_XCK,
        AUD_DACDAT,

        FPGA_I2C_SCLK
    );

input          CLOCK_50;                //    50 MHz

// Declare your inputs and outputs here
// Do not change the following outputs
output        VGA_CLK;                  //    VGA Clock
output        VGA_HS;                   //    VGA
H_SYNC
output        VGA_VS;                   //    VGA
V_SYNC
output        VGA_BLANK_N;              //    VGA BLANK
output        VGA_SYNC_N;              //    VGA SYNC
output [9:0]   VGA_R;                   //    VGA Red[9:0]
output [9:0]   VGA_G;                   //    VGA Green[9:0]

```

```

output [9:0]  VGA_B;                                //    VGA Blue[9:0]

input  [3:0] KEY;
input  [9:0] SW;
output      [9:0] LEDR;
output      [6:0] HEX0;
output      [6:0] HEX1;
output      [6:0] HEX2;

input      PS2_DAT; // PS2 data line
input      PS2_CLK; // PS2 clock line


input      AUD_ADCDAT;

// Bidirectionals
inout      AUD_BCLK;
inout      AUD_ADCLRCK;
inout      AUD_DACLK;

inout      FPGA_I2C_SDAT;

// Outputs
output      AUD_XCK;
output      AUD_DACDAT;

output      FPGA_I2C_SCLK;

```

```

// Internal Wires

wire          audio_in_available;
wire [31:0] left_channel_audio_in;
wire [31:0] right_channel_audio_in;
wire          read_audio_in;


wire          audio_out_allowed;
wire [31:0] left_channel_audio_out;
wire [31:0] right_channel_audio_out;
wire          write_audio_out;


// Internal Registers

reg [18:0] delay_cnt;
wire [18:0] delay;


reg snd;


wire [11:0] gameRAM_DataOut_Display;


// Create the colour, x, y and writeEn wires that are inputs to the controller.

wire [2:0] colour;
wire [8:0] x;
wire [7:0] y;
wire writeEn;
wire [3:0] sig_move;

```

```

wire resetn;

wire enter;

wire colourful;

// Create an Instance of a VGA controller - there can be only one!
// Define the number of colours as well as the initial background
// image file (.MIF) for the controller.
vga_adapter VGA(
    .resetn(resetn),
    .clock(CLOCK_50),
    .colour(colour),
    .x(x),
    .y(y),
    .plot(writeEn),
    /* Signals for the DAC to drive the monitor. */
    .VGA_R(VGA_R),
    .VGA_G(VGA_G),
    .VGA_B(VGA_B),
    .VGA_HS(VGA_HS),
    .VGA_VS(VGA_VS),
    .VGA_BLANK(VGA_BLANK_N),
    .VGA_SYNC(VGA_SYNC_N),
    .VGA_CLK(VGA_CLK));

defparam VGA.RESOLUTION = "320x240";
defparam VGA.MONOCHROME = "FALSE";
defparam VGA.BITS_PER_COLOUR_CHANNEL = 1;
defparam VGA.BACKGROUND_IMAGE = "titlescreen.mif";

```

// Put your code here. Your code should produce signals x,y,colour and writeEn  
// for the VGA controller, in addition to any other functionality your design may  
require.

```
wire dummy_valid;
wire dummy_makeBreak;
wire [7:0] dummy_outCode;
wire [3:0] keyboard_sig_move;
wire keyboard_reset;
wire keyboard_enter;

keyboard_press_driver u0(
    .CLOCK_50(CLOCK_50),
    .valid(dummy_valid),
    .makeBreak(dummy_makeBreak),
    .outCode(dummy_outCode),
    .KEYBOARD_RESET(keyboard_reset),
    .KEYBOARD_ENTER(keyboard_enter),
    .sig_move(keyboard_sig_move),////////////////////////////////////
    .PS2_DAT(PS2_DAT),
    .PS2_CLK(PS2_CLK),
    .reset(~resetn)
);

wire [4:0] randMove;
wire CLOCK_120HZ;

RateDivider_120HZ rd_120_1(
    .CLOCK_50(CLOCK_50),
```

```

        .resetn(resetn),
        .CLOCK_120HZ(CLOCK_120HZ)
    );

```

```

Linear_FB_Shift_Reg_5b randMoveGen(
    .CLOCK_50(CLOCK_50),
    .resetn(resetn),
    .LFBSR_enable(CLOCK_120HZ),
    .out(randMove)
);

```

```

reg up;
reg down;
reg left;
reg right;

```

```

always @ (*) begin
    up = 1'b0;
    down = 1'b0;
    left = 1'b0;
    right = 1'b0;
    case (randMove[4:1])
        4'b0000:
            up = 1'b1 & CLOCK_120HZ;
        4'b0001:
            up = 1'b1 & CLOCK_120HZ;
        4'b0010:
            up = 1'b1 & CLOCK_120HZ;

```



```

4'b0011:
    up = 1'b1 & CLOCK_120HZ;
4'b0100:
    down = 1'b1 & CLOCK_120HZ;
4'b0101:
    down = 1'b1 & CLOCK_120HZ;
4'b0110:
    down = 1'b1 & CLOCK_120HZ;
4'b0111:
    down = 1'b1 & CLOCK_120HZ;
4'b1000:
    left = 1'b1 & CLOCK_120HZ;
4'b1001:
    left = 1'b1 & CLOCK_120HZ;
4'b1010:
    left = 1'b1 & CLOCK_120HZ;
4'b1011:
    left = 1'b1 & CLOCK_120HZ;
4'b1100:
    right = 1'b1 & CLOCK_120HZ;
4'b1101:
    right = 1'b1 & CLOCK_120HZ;
4'b1110:
    right = 1'b1 & CLOCK_120HZ;
4'b1111:
    right = 1'b1 & CLOCK_120HZ;
endcase
end

```

```

// Controls for 2048
// KEY[3] Left
// KEY[2] Up
// KEY[1] Down
// KEY[0] Right
    assign sig_move = SW[6] ? {left, up, down, right} : ({~KEY[3], ~KEY[2], ~KEY[1],
~KEY[0]} | keyboard_sig_move);

    assign resetn = ((SW[9]) | (keyboard_reset));

    assign enter = ((SW[8]) | (keyboard_enter));

    assign colourful = SW[7];

    handler handler0(
        .CLOCK_50(CLOCK_50),
        .resetn(resetn),

        .sig_move(sig_move),
        .enter(enter),

        .gameRAM_Addr_Display(SW[3:0]),
        .gameRAM_DataOut_Display(gameRAM_DataOut_Display),

        .stateLEDs(LED_R),

        .x(x),

```

```
.y(y),  
.colour(colour),  
.writeEn(writeEn),
```

```
.colourful(colourful)  
);
```

```
hex_decoder h0(  
    .hex_digit(gameRAM_DataOut_Display[3:0]),  
    .segments(HEX0)  
);
```

```
hex_decoder h1(  
    .hex_digit(gameRAM_DataOut_Display[7:4]),  
    .segments(HEX1)  
);
```

```
hex_decoder h2(  
    .hex_digit(gameRAM_DataOut_Display[11:8]),  
    .segments(HEX2)  
);
```

```
always @(posedge CLOCK_50)  
    if(delay_cnt == delay) begin  
        delay_cnt <= 0;  
        snd <= !snd;  
    end else delay_cnt <= delay_cnt + 1;
```

```
assign delay = {sig_move, 15'd3000};
```

```
wire [31:0] sound = (SW == 0) ? 0 : snd ? 32'd10000000 : -32'd10000000;
```

```
assign read_audio_in          = audio_in_available &  
audio_out_allowed;
```

```
assign left_channel_audio_out  = left_channel_audio_in+sound;  
assign right_channel_audio_out = right_channel_audio_in+sound;  
assign write_audio_out        = audio_in_available &  
audio_out_allowed;
```

```
Audio_Controller Audio_Controller (
```

```
    // Inputs
```

```
    .CLOCK_50                      (CLOCK_50),
```

```
    .reset                         (~resetn),
```

```
    .clear_audio_in_memory        (),
```

```
    .read_audio_in                (read_audio_in),
```

```
    .clear_audio_out_memory       (),
```

```
    .left_channel_audio_out       (left_channel_audio_out),
```

```
    .right_channel_audio_out      (right_channel_audio_out),
```

```
    .write_audio_out              (write_audio_out),
```

```
    .AUD_ADCCDAT                  (AUD_ADCCDAT),
```

```

// Bidirectionals
.AUD_BCLK                      (AUD_BCLK),
.AUD_ADCLRCK                   (AUD_ADCLRCK),
.AUD_DACLCK                    (AUD_DACLCK),

// Outputs
.audio_in_available            (audio_in_available),
.left_channel_audio_in         (left_channel_audio_in),
.right_channel_audio_in        (right_channel_audio_in),

.audio_out_allowed             (audio_out_allowed),

.AUD_XCK                       (AUD_XCK),
.AUD_DACDAT                    (AUD_DACDAT)

);

avconf #(.USE_MIC_INPUT(1)) avc (
    .FPGA_I2C_SCLK              (FPGA_I2C_SCLK),
    .FPGA_I2C_SDAT              (FPGA_I2C_SDAT),
    .CLOCK_50                   (CLOCK_50),
    .reset                      (~resetn)
);

endmodule

module handler(

```

```

input CLOCK_50,
input resetn,

input [3:0] sig_move,
input enter,

input [3:0] gameRAM_Addr_Display,
output [11:0] gameRAM_DataOut_Display,

output [9:0] stateLEDs,

output [8:0] x,
output [7:0] y,
output [2:0] colour,
output writeEn,

input colourful
);

wire CLOCK_60HZ;

RateDivider_60HZ rd_60(
    .CLOCK_50(CLOCK_50),
    .resetn(resetn),
    .CLOCK_60HZ(CLOCK_60HZ)
);

wire CLOCK_120HZ;

```

```

RateDivider_120HZ rd_30(
    .CLOCK_50(CLOCK_50),
    .resetn(resetn),
    .CLOCK_120HZ(CLOCK_120HZ)
);

wire [11:0] gameRAM_DataIn;
wire [11:0] gameRAM_DataOut;
wire [3:0] gameRAM_Addr;
wire gameRAM_writeEn;

wire [11:0] gameRAM_DataIn_Dummy;
wire gameRAM_writeEn_Dummy;

assign gameRAM_DataIn_Dummy = 12'b0;
assign gameRAM_writeEn_Dummy = 1'b0;

GameRAM2port gameBoard(
    .clock(CLOCK_50),
    .data_a(gameRAM_DataIn),
    .address_a(gameRAM_Addr),
    .wren_a(gameRAM_writeEn),
    .q_a(gameRAM_DataOut),
    .data_b(gameRAM_DataIn_Dummy),
    .address_b(gameRAM_Addr_Display),
    .wren_b(gameRAM_writeEn_Dummy),
    .q_b(gameRAM_DataOut_Display)

```

);

```
wire sig_clearBoard_DONE;  
wire sig_randNum_GOOD;  
wire sig_drawBoard_DONE;  
wire sig_doneProcess;  
wire sig_toNoMove;  
wire sig_toMergeMove;  
wire sig_toJustMove;  
wire sig_nextIteration;  
wire sig_clearBoard;  
wire sig_checkRandNum;  
wire sig_spawnNumOnBoard;  
wire sig_drawBoard;  
wire sig_initDraw;  
wire sig_gameDraw;  
wire sig_gameEndDraw;  
wire sig_resetIteration;  
wire sig_iterationCheck;  
wire sig_setCurrentPOS;  
wire sig_setCurrentNextPOS;  
wire sig_checkBound;  
wire sig_calcMove;  
wire sig_noMove;  
wire sig_mergeUpdateNext;  
wire sig_mergeUpdateCur;  
wire sig_noMergeUpdateNext;  
wire sig_noMergeUpdateCur;
```





```

wire [4:0] randomNum;
wire LFBSR_enable;


wire sig_cascCounter_init;
wire ld_cascCounter;
wire sig_casc_CounterCheck;
wire sig_pixelCounter_init;
wire sig_pixel_CounterEn;
wire sig_drawRandNum;
wire sel_randNum_XY;
wire sel_randNum_Colour;
wire sig_cascCounter_Incre;
wire sig_doneCasc;
wire sig_randNumDraw_DONE;


wire [6:0] casc_Counter, temp_casc_Counter;


wire [5:0] rand_eff_X;
wire [5:0] rand_eff_Y;


wire sig_getHighscore;
wire ld_highscore;


wire sig_gameLose;


wire sig_drawEnd;


control control0(

```

```

// Standard I/O
.CLOCK_50(CLOCK_50),
.resetn(resetn),

// Game control inputs
.sig_move(sig_move),
.enter(enter),

// State LEDs for debugging
.stateLEDs(stateLEDs),

// Signals from datapath
.sig_clearBoard_DONE(sig_clearBoard_DONE),
.sig_randNum_GOOD(sig_randNum_GOOD),
.sig_drawBoard_DONE(sig_drawBoard_DONE),
.sig_doneProcess(sig_doneProcess),
.sig_toNoMove(sig_toNoMove),
.sig_toMergeMove(sig_toMergeMove),
.sig_toJustMove(sig_toJustMove),
.sig_nextIteration(sig_nextIteration),
.sig_debug_displayBoard_DONE(sig_debug_displayBoard_DONE),//////////
//////////
.sig_drawBoard_Cont(sig_drawBoard_Cont),

// Signals to datapath
.sig_clearBoard(sig_clearBoard),
.sig_checkRandNum(sig_checkRandNum),
.sig_spawnNumOnBoard(sig_spawnNumOnBoard),
.sig_drawBoard(sig_drawBoard),

```

```

.sig_drawBoard_init(sig_drawBoard_init),////////////////////
.sig_initDraw(sig_initDraw),
.sig_gameDraw(sig_gameDraw),
.sig_gameEndDraw(sig_gameEndDraw),
.sig_resetIteration(sig_resetIteration),
.sig_iterationCheck(sig_iterationCheck),
.sig_setCurrentPOS(sig_setCurrentPOS),
.sig_setCurrentNextPOS(sig_setCurrentNextPOS),
.sig_checkBound(sig_checkBound),
.sig_calcMove(sig_calcMove),
.sig_noMove(sig_noMove),
.sig_IdExt(sig_IdExt), //////////////////////////////////////
.sig_mergeUpdateNext(sig_mergeUpdateNext),
.sig_mergeUpdateCur(sig_mergeUpdateCur),
.sig_noMergeUpdateNext(sig_noMergeUpdateNext),
.sig_noMergeUpdateCur(sig_noMergeUpdateCur),
.sig_iterationIncre(sig_iterationIncre),
.sig_debug_displayBoard(sig_debug_displayBoard),////////////////////
////////////////////////////////////
.sig_drawBoard_CounterCheck(sig_drawBoard_CounterCheck),
.sig_getCur_XY(sig_getCur_XY),
.sig_drawBoard_CounterEn(sig_drawBoard_CounterEn),
.gameRAM_writeEn(gameRAM_writeEn),
.Id_randomNum(Id_randomNum),
.Id_move(Id_move),
.Id_iterationCounter(Id_iterationCounter),
.Id_gameBoard_cur_X(Id_gameBoard_cur_X),
.Id_gameBoard_cur_Y(Id_gameBoard_cur_Y),
.Id_gameBoard_cur_Value(Id_gameBoard_cur_Value),

```

```

.Id_gameBoard_next_X(Id_gameBoard_next_X),
.Id_gameBoard_next_Y(Id_gameBoard_next_Y),
.writeEn(writeEn),
.LFBSR_enable(LFBSR_enable),

.sig_cascCounter_init(sig_cascCounter_init),
.Id_cascCounter(Id_cascCounter),
.sig_casc_CounterCheck(sig_casc_CounterCheck),
.sig_pixelCounter_init(sig_pixelCounter_init),
.sig_pixel_CounterEn(sig_pixel_CounterEn),
.sig_drawRandNum(sig_drawRandNum),
.sel_randNum_XY(sel_randNum_XY),
.sel_randNum_Colour(sel_randNum_Colour),
.sig_cascCounter_Incre(sig_cascCounter_Incre),
.sig_doneCasc(sig_doneCasc),
.sig_randNumDraw_DONE(sig_randNumDraw_DONE),
.CLOCK_60HZ(CLOCK_60HZ),

.sig_getHighscore(sig_getHighscore),
.Id_highscore(Id_highscore),

.sig_gameLose(sig_gameLose),

.sig_drawEnd(sig_drawEnd),

.CLOCK_120HZ(CLOCK_120HZ)
);

```

```

datapath datapath0(
// Standard I/O
.CLOCK_50(CLOCK_50),
.resetn(resetn),

.sig_move(sig_move),

// Signals from control
.sig_clearBoard(sig_clearBoard),
.sig_checkRandNum(sig_checkRandNum),
.sig_spawnNumOnBoard(sig_spawnNumOnBoard),
.sig_drawBoard(sig_drawBoard),
.sig_drawBoard_init(sig_drawBoard_init),////////////////////
.sig_initDraw(sig_initDraw),
.sig_gameDraw(sig_gameDraw),
.sig_gameEndDraw(sig_gameEndDraw),
.sig_resetIteration(sig_resetIteration),
.sig_iterationCheck(sig_iterationCheck),
.sig_setCurrentPOS(sig_setCurrentPOS),
.sig_setCurrentNextPOS(sig_setCurrentNextPOS),
.sig_checkBound(sig_checkBound),
.sig_calcMove(sig_calcMove),
.sig_noMove(sig_noMove),
.sig_IdExt(sig_IdExt), //////////////////////
.sig_mergeUpdateNext(sig_mergeUpdateNext),
.sig_mergeUpdateCur(sig_mergeUpdateCur),
.sig_noMergeUpdateNext(sig_noMergeUpdateNext),
.sig_noMergeUpdateCur(sig_noMergeUpdateCur),

```

```

.sig_iterationIncre(sig_iterationIncre),
.sig_debug_displayBoard(sig_debug_displayBoard),////////////////////////////////////
////////////////////////////////////
.sig_drawBoard_CounterCheck(sig_drawBoard_CounterCheck),
.sig_getCur_XY(sig_getCur_XY),
.sig_drawBoard_CounterEn(sig_drawBoard_CounterEn),
.gameRAM_writeEn(gameRAM_writeEn),
.Id_randomNum(Id_randomNum),
.Id_move(Id_move),
.Id_iterationCounter(Id_iterationCounter),
.Id_gameBoard_cur_X(Id_gameBoard_cur_X),
.Id_gameBoard_cur_Y(Id_gameBoard_cur_Y),
.Id_gameBoard_cur_Value(Id_gameBoard_cur_Value),
.Id_gameBoard_next_X(Id_gameBoard_next_X),
.Id_gameBoard_next_Y(Id_gameBoard_next_Y),
.LFBSR_enable(LFBSR_enable),

// Signals to control
.sig_clearBoard_DONE(sig_clearBoard_DONE),
.sig_randNum_GOOD(sig_randNum_GOOD),
.sig_drawBoard_DONE(sig_drawBoard_DONE),
.sig_doneProcess(sig_doneProcess),
.sig_toNoMove(sig_toNoMove),
.sig_toMergeMove(sig_toMergeMove),
.sig_toJustMove(sig_toJustMove),
.sig_nextIteration(sig_nextIteration),
.sig_debug_displayBoard_DONE(sig_debug_displayBoard_DONE),////////////////////////////////////
////////////////////////////////////
.sig_drawBoard_Cont(sig_drawBoard_Cont),

```

```

// Game RAM I/O
.gameRAM_DataOut(gameRAM_DataOut),
.gameRAM_DataIn(gameRAM_DataIn),
.gameRAM_Addr(gameRAM_Addr),

// VGA output
.x(x),
.y(y),
.colour(colour),

.effective_X(effective_X),
.effective_Y(effective_Y),

.rand_eff_X(rand_eff_X),
.rand_eff_Y(rand_eff_Y),

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
.gameBoard_cur_X(cur_X),
.gameBoard_next_X(next_X),
.gameBoard_cur_Y(cur_Y),
.gameBoard_next_Y(next_Y),
.gameBoard_cur_Value(cur_Value),
.temp_X(temp_X),
.temp_Y(temp_Y),

.casc_Counter(casc_Counter),
.temp_casc_Counter(temp_casc_Counter),

```



```

.randomNum_reg(randomNum_reg),
.randomNum(randomNum),

.sig_cascCounter_init(sig_cascCounter_init),
.Id_cascCounter(Id_cascCounter),
.sig_casc_CounterCheck(sig_casc_CounterCheck),
.sig_pixelCounter_init(sig_pixelCounter_init),
.sig_pixel_CounterEn(sig_pixel_CounterEn),
.sig_drawRandNum(sig_drawRandNum),
.sel_randNum_XY(sel_randNum_XY),
.sel_randNum_Colour(sel_randNum_Colour),
.sig_cascCounter_Incre(sig_cascCounter_Incre),
.sig_doneCasc(sig_doneCasc),
.sig_randNumDraw_DONE(sig_randNumDraw_DONE),

.sig_getHighscore(sig_getHighscore),
.Id_highscore(Id_highscore),

.sig_gameLose(sig_gameLose),

.sig_drawEnd(sig_drawEnd),

.colourful(colourful)
);

```

```
endmodule
```

```

module RateDivider_60HZ(CLOCK_50, resetn, CLOCK_60HZ);

    input CLOCK_50, resetn;
    output CLOCK_60HZ;

    reg CLOCK_60HZ;
    reg [21:0] counter;

    always @(posedge CLOCK_50, negedge resetn) begin
        if (!resetn) begin
            counter <= 22'd0;
            CLOCK_60HZ <= 1'b0;
        end
        else begin
            counter <= (counter == 22'd2499999) ? 22'd0 : counter + 1'b1;
            CLOCK_60HZ <= (counter == 22'd833333) | (counter ==
22'd1666666) | (counter == 22'd2499999);
        end
    end

endmodule

```

```

module RateDivider_120HZ(CLOCK_50, resetn, CLOCK_120HZ);

```

```

    input CLOCK_50, resetn;
    output CLOCK_120HZ;

```

```

    reg CLOCK_120HZ;
    reg [20:0] counter;

```

```

always @(posedge CLOCK_50, negedge resetn) begin
    if (!resetn) begin
        counter <= 21'd0;
        CLOCK_120HZ <= 1'b0;
    end
    else begin
        counter <= (counter == 21'd1249999) ? 21'd0 : counter + 1'b1;
        CLOCK_120HZ <= (counter == 21'd416666) | (counter ==
21'd833333) | (counter == 21'd1249999);
    end
end
end

```

```

endmodule

```

```

module control(
    // Standard I/O
    input CLOCK_50,
    input resetn,

    // Game control inputs
    input [3:0] sig_move,
    input enter,

    // State LEDs for debugging
    output [9:0] stateLEDs,

    // Signals from datapath
    input sig_clearBoard_DONE,

```

```

input sig_randNum_GOOD,
input sig_drawBoard_DONE,
input sig_doneProcess,
input sig_toNoMove,
input sig_toMergeMove,
input sig_toJustMove,
input sig_nextIteration,
input sig_debug_displayBoard_DONE,
////////////////////////////////////
input sig_drawBoard_Cont,

// Signals to datapath
output reg sig_clearBoard,
output reg sig_checkRandNum,
output reg sig_spawnNumOnBoard,
output reg sig_drawBoard,
output reg sig_drawBoard_init,////////////////////////////////////
output reg sig_initDraw,
output reg sig_gameDraw,
output reg sig_gameEndDraw,
output reg sig_resetIteration,
output reg sig_iterationCheck,
output reg sig_setCurrentPOS,
output reg sig_setCurrentNextPOS,
output reg sig_checkBound,
output reg sig_calcMove,
output reg sig_noMove,
output reg sig_IdExt, //////////////////////////////////////
output reg sig_mergeUpdateNext,

```

```

output reg sig_mergeUpdateCur,
output reg sig_noMergeUpdateNext,
output reg sig_noMergeUpdateCur,
output reg sig_iterationIncre,
output reg sig_debug_displayBoard, //////////////////////////////////////
output reg sig_drawBoard_CounterCheck,
output reg sig_getCur_XY,
output reg sig_drawBoard_CounterEn,
output reg gameRAM_writeEn,
output reg ld_randomNum,
output reg ld_move,
output reg ld_iterationCounter,
output reg ld_gameBoard_cur_X,
output reg ld_gameBoard_cur_Y,
output reg ld_gameBoard_cur_Value,
output reg ld_gameBoard_next_X,
output reg ld_gameBoard_next_Y,
output reg LFBSR_enable,
output reg writeEn,

input sig_doneCasc,
input sig_randNumDraw_DONE,
input CLOCK_60HZ,

output reg sig_cascCounter_init,
output reg ld_cascCounter,
output reg sig_casc_CounterCheck,
output reg sig_pixelCounter_init,

```

```

output reg sig_pixel_CounterEn,
output reg sig_drawRandNum,
output reg sel_randNum_XY,
output reg sel_randNum_Colour,
output reg sig_cascCounter_Incre,

```

```

output reg sig_getHighscore,
output reg ld_highscore,

```

```

input sig_gameLose,

```

```

output reg sig_drawEnd,

```

```

input CLOCK_120HZ

```

```

);

```

```

reg [6:0] current_state, next_state;

```

```

localparam TITLE_SCREEN
= 7'd1,

```

```

                                TITLE_SCREEN_WAIT
                                = 7'd2,

```

```

                                INIT_CLEAR_BOARD
                                = 7'd3,

```

```

                                INIT_RAND_NUM
                                = 7'd4,

```

```

                                INIT_CHECK_NUM
                                = 7'd5,

```

```

                                INIT_SPAWN_NUM
                                = 7'd6,

```

	INIT_DRAW_INIT
= 7'd7,//////////	
	INIT_DRAW_COUNTER_CHECK
= 7'd8,	
	INIT_DRAW_LD_XY
= 7'd9,	
	INIT_DRAW_LD_VAL
= 7'd10,	
	INIT_DRAW_LD_VAL_2
= 7'd11,	
	INIT_DRAW
= 7'd12,	
	INIT_DRAW_COUNTER_INCRE
= 7'd13,	
	GAME_WAIT_FOR_MOVE
= 7'd14,	
	GAME_STORE_MOVE
= 7'd15,	
	GAME_STORE_MOVE_WAIT
= 7'd16,	
	GAME_PROCESS_MOVE_INIT
= 7'd17,	
	GAME_PROCESS_ITER_CHECK
= 7'd18,	
	GAME_PROCESS_SET_POS
= 7'd19,	
	GAME_SET_POS_WAIT
= 7'd20,	
	GAME_SET_POS_WAIT_2
= 7'd21,	
	GAME_PROCESS_WITHIN_BOUND
= 7'd22,	
	GAME_PROCESS_SET_NEXT_POS
= 7'd23,	

```

                                GAME_PROCESS_CALC_PRE
= 7'd24,

                                GAME_PROCESS_CALC
                                = 7'd25,

                                GAME_PROCESS_NO_MOVE
                                = 7'd26,

                                NO_MOVE_LD_EXT
                                = 7'd27,

                                NO_MOVE_LD_EXT_2
                                = 7'd28,

                                GAME_PROCESS_MERGE_MOVE_UPDATE_NEXT
= 7'd29,

GAME_PROCESS_MERGE_MOVE_UPDATE_NEXT_EXT= 7'd30,

                                GAME_PROCESS_MERGE_MOVE_UPDATE_CUR
= 7'd31,

GAME_PROCESS_MERGE_MOVE_UPDATE_CUR_EXT  = 7'd32,

                                GAME_PROCESS_JUST_MOVE_UPDATE_NEXT
= 7'd33,

GAME_PROCESS_JUST_MOVE_UPDATE_NEXT_EXT  = 7'd34,

                                GAME_PROCESS_JUST_MOVE_UPDATE_CUR
= 7'd35,

GAME_PROCESS_JUST_MOVE_UPDATE_CUR_EXT    = 7'd36,

                                GAME_PROCESS_ITER_INCRE
                                = 7'd37,

                                GAME_RAND_NUM
                                = 7'd38,

                                GAME_CHECK_NUM
                                = 7'd39,

                                GAME_CHECK_NUM_WAIT
                                = 7'd40,

```



```

        GAME_SPAWN_NUM
    = 7'd41,

        GAME_DRAW_INIT
    = 7'd42,

        GAME_DRAW_COUNTER_CHECK
    = 7'd43,

        GAME_DRAW_LD_XY
    = 7'd44,

        GAME_DRAW_LD_VAL
    = 7'd45,

        GAME_DRAW_LD_VAL_2
    = 7'd46,

        GAME_DRAW
    = 7'd47,

        GAME_DRAW_COUNTER_INCRE
    = 7'd48,

        DEBUG_DISPLAY_BOARD
    = 7'd49, //////////////////////////////////////

        DEBUG_DISPLAY_BOARD_2
    = 7'd50, //////////////////////////////////////

        CASC_COUNTER_INIT
    = 7'd51,

        CASC_COUNTER_CHECK
    = 7'd52,

        PIXEL_COUNTER_INIT
    = 7'd53,

        RAND_NUM_DRAW
    = 7'd54,

        INCRE_CASC_COUNTER
    = 7'd55,

        GAME_CASC_COUNTER_INIT
    = 7'd56,

        GAME_CASC_COUNTER_CHECK
    = 7'd57,

```

```

                                GAME_PIXEL_COUNTER_INIT
= 7'd58,

                                GAME_RAND_NUM_DRAW
                                = 7'd59,

                                GAME_INCRE_CASC_COUNTER
                                = 7'd60,

                                GAME_END_DRAW_INIT
= 7'd61,

                                GAME_END_DRAW_COUNTER_CHECK
= 7'd62,

                                GAME_END_DRAW
                                = 7'd63,

                                GAME_END_DRAW_COUNTER_INCRE
= 7'd64,

                                GAME_END
                                = 7'd65;

```

initial begin

```

current_state = TITLE_SCREEN;

sig_clearBoard                                = 1'b0;
sig_checkRandNum                              = 1'b0;
sig_spawnNumOnBoard                           = 1'b0;
sig_drawBoard                                 = 1'b0;
sig_drawBoard_init                            = 1'b0;
sig_initDraw                                  = 1'b0;
sig_gameDraw                                  = 1'b0;
sig_gameEndDraw                               = 1'b0;
sig_resetIteration                            = 1'b0;
sig_iterationCheck                            = 1'b0;
sig_setCurrentPOS                             = 1'b0;
sig_setCurrentNextPOS                         = 1'b0;

```

```

sig_checkBound                      = 1'b0;
sig_calcMove                        = 1'b0;
sig_noMove                          = 1'b0;
sig_IdExt                           =
1'b0;////////////////////////////////////

sig_mergeUpdateNext                 = 1'b0;
sig_mergeUpdateCur                 = 1'b0;
sig_noMergeUpdateNext              = 1'b0;
sig_noMergeUpdateCur              = 1'b0;
sig_iterationIncre                  = 1'b0;
sig_debug_displayBoard              = 1'b0;////////////////////////////////////
sig_drawBoard_CounterCheck         = 1'b0;
sig_getCur_XY                      = 1'b0;
sig_drawBoard_CounterEn            = 1'b0;


sig_cascCounter_init               = 1'b0;
ld_cascCounter                     = 1'b0;
sig_casc_CounterCheck              = 1'b0;
sig_pixelCounter_init              = 1'b0;
sig_pixel_CounterEn                = 1'b0;
sig_drawRandNum                    = 1'b0;
sel_randNum_XY                     = 1'b0;
sel_randNum_Colour                 = 1'b0;
sig_cascCounter_Incre              = 1'b0;


gameRAM_writeEn                    = 1'b0;
ld_randomNum                       = 1'b0;
ld_move                            = 1'b0;
ld_iterationCounter                = 1'b0;

```

```

    Id_gameBoard_cur_X                = 1'b0;
    Id_gameBoard_cur_Y                = 1'b0;
    Id_gameBoard_cur_Value            = 1'b0;
    Id_gameBoard_next_X               = 1'b0;
    Id_gameBoard_next_Y               = 1'b0;
    LFBSR_enable                      = 1'b0;
    writeEn                           = 1'b0;

    sig_getHighscore                  = 1'b0;
    Id_highscore                      = 1'b0;

    sig_drawEnd                       = 1'b0;
end

// State Table
always@(*)
begin:state_table
    case (current_state)
        // Title screen
        TITLE_SCREEN: next_state = enter ? TITLE_SCREEN_WAIT :
TITLE_SCREEN;
        TITLE_SCREEN_WAIT: next_state = ~enter ?
INIT_CLEAR_BOARD : TITLE_SCREEN_WAIT;

        // Empties board
        INIT_CLEAR_BOARD: next_state = sig_clearBoard_DONE ?
INIT_DRAW_INIT : INIT_CLEAR_BOARD;

        // Draws

```

```

INIT_DRAW_INIT: next_state = INIT_DRAW_COUNTER_CHECK;

INIT_DRAW_COUNTER_CHECK: next_state =
sig_drawBoard_Cont ? INIT_DRAW_LD_XY : INIT_RAND_NUM;

INIT_DRAW_LD_XY: next_state = INIT_DRAW_LD_VAL;

INIT_DRAW_LD_VAL: next_state = INIT_DRAW_LD_VAL_2;

INIT_DRAW_LD_VAL_2: next_state = INIT_DRAW;

INIT_DRAW: next_state = INIT_DRAW_COUNTER_INCRE;

INIT_DRAW_COUNTER_INCRE: next_state =
INIT_DRAW_COUNTER_CHECK;


// Spawns starting number

INIT_RAND_NUM: next_state = INIT_CHECK_NUM;

INIT_CHECK_NUM: next_state = sig_randNum_GOOD ?
INIT_SPAWN_NUM : INIT_RAND_NUM;//////////sig_randNum_GOOD

INIT_SPAWN_NUM: next_state = CASC_COUNTER_INIT;


// Draws random number

CASC_COUNTER_INIT: next_state = CASC_COUNTER_CHECK;

CASC_COUNTER_CHECK: next_state = sig_doneCasc ?
GAME_WAIT_FOR_MOVE : PIXEL_COUNTER_INIT;

PIXEL_COUNTER_INIT: next_state = CLOCK_120HZ ?
RAND_NUM_DRAW : PIXEL_COUNTER_INIT;

RAND_NUM_DRAW: next_state = sig_randNumDraw_DONE ?
INCRE_CASC_COUNTER : RAND_NUM_DRAW;

INCRE_CASC_COUNTER: next_state =
CASC_COUNTER_CHECK;


// Wait for moves

GAME_WAIT_FOR_MOVE: next_state = (sig_move == 4'b0) ?
GAME_WAIT_FOR_MOVE : GAME_STORE_MOVE;

```

```

// Processing

    GAME_STORE_MOVE: next_state =
GAME_STORE_MOVE_WAIT;

        GAME_STORE_MOVE_WAIT: next_state = (sig_move == 4'b0) ?
GAME_PROCESS_MOVE_INIT : GAME_STORE_MOVE_WAIT;

            GAME_PROCESS_MOVE_INIT: next_state =
GAME_PROCESS_ITER_CHECK;

                GAME_PROCESS_ITER_CHECK: next_state = sig_doneProcess
? DEBUG_DISPLAY_BOARD_2 : GAME_PROCESS_SET_POS;////changed first result

                    GAME_PROCESS_SET_POS: next_state =
GAME_SET_POS_WAIT;

                        GAME_SET_POS_WAIT: next_state =
GAME_SET_POS_WAIT_2;////////////////////////////////////

                            GAME_SET_POS_WAIT_2: next_state =
GAME_PROCESS_WITHIN_BOUND;////////////////////////////////////

                                GAME_PROCESS_WITHIN_BOUND: next_state = sig_toNoMove
? GAME_PROCESS_NO_MOVE : GAME_PROCESS_SET_NEXT_POS;

                                    GAME_PROCESS_SET_NEXT_POS: next_state =
GAME_PROCESS_CALC_PRE;

//                                DEBUG_SET_NEXT_POS_WAIT: next_state =
GAME_PROCESS_CALC;

                                    GAME_PROCESS_CALC_PRE: next_state =
GAME_PROCESS_CALC;

                                        GAME_PROCESS_CALC: next_state = sig_toNoMove ?
GAME_PROCESS_NO_MOVE : (sig_toMergeMove ?
GAME_PROCESS_MERGE_MOVE_UPDATE_NEXT : (sig_toJustMove ?
GAME_PROCESS_JUST_MOVE_UPDATE_NEXT :
GAME_PROCESS_JUST_MOVE_UPDATE_NEXT));

                                            GAME_PROCESS_NO_MOVE: next_state = sig_nextIteration ?
GAME_PROCESS_ITER_INCRE : NO_MOVE_LD_EXT;

                                                NO_MOVE_LD_EXT: next_state = NO_MOVE_LD_EXT_2;

                                                    NO_MOVE_LD_EXT_2: next_state =
GAME_PROCESS_WITHIN_BOUND;

//                                DEBUG_WAIT_NO_MOVE:

```

```

        GAME_PROCESS_MERGE_MOVE_UPDATE_NEXT: next_state =
        GAME_PROCESS_MERGE_MOVE_UPDATE_NEXT_EXT;

        GAME_PROCESS_MERGE_MOVE_UPDATE_NEXT_EXT:
next_state = GAME_PROCESS_MERGE_MOVE_UPDATE_CUR;

        GAME_PROCESS_MERGE_MOVE_UPDATE_CUR: next_state =
        GAME_PROCESS_MERGE_MOVE_UPDATE_CUR_EXT;

        GAME_PROCESS_MERGE_MOVE_UPDATE_CUR_EXT:
next_state = GAME_PROCESS_NO_MOVE;

        GAME_PROCESS_JUST_MOVE_UPDATE_NEXT: next_state =
        GAME_PROCESS_JUST_MOVE_UPDATE_NEXT_EXT;

        GAME_PROCESS_JUST_MOVE_UPDATE_NEXT_EXT:
next_state = GAME_PROCESS_JUST_MOVE_UPDATE_CUR;

        GAME_PROCESS_JUST_MOVE_UPDATE_CUR: next_state =
        GAME_PROCESS_JUST_MOVE_UPDATE_CUR_EXT;

        GAME_PROCESS_JUST_MOVE_UPDATE_CUR_EXT: next_state
= GAME_PROCESS_NO_MOVE;

        GAME_PROCESS_ITER_INCRE: next_state =
        GAME_PROCESS_ITER_CHECK;


        // Debug display RAM

        DEBUG_DISPLAY_BOARD_2: next_state =
sig_debug_displayBoard_DONE ? GAME_DRAW_INIT :
        DEBUG_DISPLAY_BOARD_2; ///$#####

        // Spawn new number

        GAME_RAND_NUM: next_state = GAME_CHECK_NUM;

        GAME_CHECK_NUM: next_state = GAME_CHECK_NUM_WAIT;

        GAME_CHECK_NUM_WAIT: next_state = sig_randNum_GOOD ?
        GAME_SPAWN_NUM : (sig_gameLose ? GAME_END_DRAW_INIT :
        GAME_RAND_NUM); ////////////////sig_randNum_GOOD

        GAME_SPAWN_NUM: next_state =
        GAME_CASC_COUNTER_INIT;

```

```

// Debug display RAM

//          DEBUG_DISPLAY_BOARD: next_state =
sig_debug_displayBoard_DONE ? GAME_DRAW_INIT : DEBUG_DISPLAY_BOARD;
////

// Draw new board

GAME_DRAW_INIT: next_state =
GAME_DRAW_COUNTER_CHECK;

GAME_DRAW_COUNTER_CHECK: next_state =
sig_drawBoard_Cont ? GAME_DRAW_LD_XY :
GAME_RAND_NUM;/$#####

GAME_DRAW_LD_XY: next_state = GAME_DRAW_LD_VAL;
GAME_DRAW_LD_VAL: next_state = GAME_DRAW_LD_VAL_2;
GAME_DRAW_LD_VAL_2: next_state = GAME_DRAW;
GAME_DRAW: next_state = GAME_DRAW_COUNTER_INCRE;

GAME_DRAW_COUNTER_INCRE: next_state =
GAME_DRAW_COUNTER_CHECK;

//          GAME_DRAW: next_state = sig_drawBoard_DONE ?
GAME_WAIT_FOR_MOVE : GAME_DRAW;

// Draws random number

GAME_CASC_COUNTER_INIT: next_state =
GAME_CASC_COUNTER_CHECK;

GAME_CASC_COUNTER_CHECK: next_state = sig_doneCasc ?
GAME_WAIT_FOR_MOVE : GAME_PIXEL_COUNTER_INIT;

GAME_PIXEL_COUNTER_INIT: next_state = CLOCK_120HZ ?
GAME_RAND_NUM_DRAW : GAME_PIXEL_COUNTER_INIT;

GAME_RAND_NUM_DRAW: next_state =
sig_randNumDraw_DONE ? GAME_INCRE_CASC_COUNTER :
GAME_RAND_NUM_DRAW;

GAME_INCRE_CASC_COUNTER: next_state =
GAME_CASC_COUNTER_CHECK;

```



```

        // Game end sequence

        GAME_END_DRAW_INIT: next_state =
GAME_END_DRAW_COUNTER_CHECK;

        GAME_END_DRAW_COUNTER_CHECK: next_state =
sig_drawBoard_Cont ? GAME_END_DRAW : GAME_END;

        GAME_END_DRAW: next_state =
GAME_END_DRAW_COUNTER_INCRE;

        GAME_END_DRAW_COUNTER_INCRE: next_state =
GAME_END_DRAW_COUNTER_CHECK;

        GAME_END: next_state = enter ? INIT_CLEAR_BOARD :
GAME_END;

        default:next_state = INIT_CLEAR_BOARD;

    endcase

end // state_table

always @(*)
begin: enable_signals

    sig_clearBoard                = 1'b0;
    sig_checkRandNum              = 1'b0;
    sig_spawnNumOnBoard           = 1'b0;
    sig_drawBoard                 = 1'b0;
    sig_drawBoard_init            = 1'b0;
    sig_initDraw                  = 1'b0;
    sig_gameDraw                  = 1'b0;
    sig_gameEndDraw               = 1'b0;
    sig_resetIteration            = 1'b0;
    sig_iterationCheck            = 1'b0;
    sig_setCurrentPOS             = 1'b0;
    sig_setCurrentNextPOS        = 1'b0;

```

```

sig_checkBound                      = 1'b0;
sig_calcMove                        = 1'b0;
sig_noMove                          = 1'b0;
sig_IdExt                           =
1'b0;////////////////////////////////////

sig_mergeUpdateNext                  = 1'b0;
sig_mergeUpdateCur                  = 1'b0;
sig_noMergeUpdateNext                = 1'b0;
sig_noMergeUpdateCur                = 1'b0;
sig_iterationIncre                   = 1'b0;
sig_debug_displayBoard               = 1'b0;////////////////////////////////////
sig_drawBoard_CounterCheck           = 1'b0;
sig_getCur_XY                       = 1'b0;
sig_drawBoard_CounterEn              = 1'b0;


sig_cascCounter_init                 = 1'b0;
ld_cascCounter                       = 1'b0;
sig_casc_CounterCheck                = 1'b0;
sig_pixelCounter_init                = 1'b0;
sig_pixel_CounterEn                  = 1'b0;
sig_drawRandNum                      = 1'b0;
sel_randNum_XY                       = 1'b0;
sel_randNum_Colour                   = 1'b0;
sig_cascCounter_Incre                 = 1'b0;


gameRAM_writeEn                      = 1'b0;
ld_randomNum                         = 1'b0;
ld_move                             = 1'b0;
ld_iterationCounter                  = 1'b0;

```

```

ld_gameBoard_cur_X          = 1'b0;
ld_gameBoard_cur_Y          = 1'b0;
ld_gameBoard_cur_Value      = 1'b0;
ld_gameBoard_next_X         = 1'b0;
ld_gameBoard_next_Y         = 1'b0;
LFBSR_enable                = 1'b0;
writeEn                     = 1'b0;

sig_getHighscore            = 1'b0;
ld_highscore                = 1'b0;

sig_drawEnd                 = 1'b0;
case (current_state)
    GAME_END_DRAW_INIT: begin
        sig_drawBoard_init    = 1'b1;
    end
    GAME_END_DRAW_COUNTER_CHECK: begin
        sig_drawBoard_CounterCheck = 1'b1;
    end
    GAME_END_DRAW: begin
        sig_drawEnd            = 1'b1;
        writeEn                = 1'b1;
    end
    GAME_END_DRAW_COUNTER_INCRE: begin
        sig_drawBoard_CounterEn = 1'b1;
    end
    GAME_INCRE_CASC_COUNTER: begin
        sig_cascCounter_Incre   = 1'b1;
    end
end

```

```

        ld_cascCounter          = 1'b1;
end
GAME_RAND_NUM_DRAW: begin
    sig_drawRandNum            = 1'b1;
    sig_pixel_CounterEn        = 1'b1;
    sel_randNum_XY              = 1'b1;
    sel_randNum_Colour          = 1'b1;
    writeEn                     = 1'b1;
end
GAME_PIXEL_COUNTER_INIT: begin
    sig_pixelCounter_init       = 1'b1;
end
GAME_CASC_COUNTER_CHECK: begin
    sig_casc_CounterCheck       = 1'b1;
end
GAME_CASC_COUNTER_INIT: begin
    sig_cascCounter_init         = 1'b1;
    ld_cascCounter                = 1'b1;
end
INCRE_CASC_COUNTER: begin
    sig_cascCounter_Incre        = 1'b1;
    ld_cascCounter                = 1'b1;
end
RAND_NUM_DRAW: begin
    sig_drawRandNum            = 1'b1;
    sig_pixel_CounterEn        = 1'b1;
    sel_randNum_XY              = 1'b1;
    sel_randNum_Colour          = 1'b1;

```

```

        writeEn                                = 1'b1;
    end
    PIXEL_COUNTER_INIT: begin
        sig_pixelCounter_init    = 1'b1;
    end
    CASC_COUNTER_CHECK: begin
        sig_casc_CounterCheck    = 1'b1;
    end
    CASC_COUNTER_INIT: begin
        sig_cascCounter_init      = 1'b1;
        ld_cascCounter            = 1'b1;
    end
    INIT_CLEAR_BOARD: begin
        sig_clearBoard            = 1'b1;
        gameRAM_writeEn           = 1'b1;
        ld_highscore              = 1'b1;
    end
    INIT_RAND_NUM: begin
        LFBSR_enable              = 1'b1;
        ld_randomNum              = 1'b1;
    end
    INIT_CHECK_NUM: begin
        sig_checkRandNum          = 1'b1;
    end
    INIT_SPAWN_NUM: begin
        sig_spawnNumOnBoard       = 1'b1;
        gameRAM_writeEn           = 1'b1;
    end
end

```

```

INIT_DRAW_INIT: begin
    sig_drawBoard_init      = 1'b1;
end

INIT_DRAW_COUNTER_CHECK: begin
    sig_drawBoard_CounterCheck = 1'b1;
end

INIT_DRAW_LD_XY: begin
    sig_getCur_XY          = 1'b1;
    ld_gameBoard_cur_X      = 1'b1;
    ld_gameBoard_cur_Y      = 1'b1;
end

INIT_DRAW_LD_VAL: begin
    sig_ldExt                = 1'b1;
    ld_gameBoard_cur_Value = 1'b1;
end

INIT_DRAW_LD_VAL_2: begin
    sig_ldExt                = 1'b1;
    ld_gameBoard_cur_Value = 1'b1;
end

INIT_DRAW: begin
    sig_drawBoard            = 1'b1;
    writeEn                  = 1'b1;
end

INIT_DRAW_COUNTER_INCRE: begin
    sig_drawBoard_CounterEn = 1'b1;
end

GAME_STORE_MOVE: begin
    ld_move                  = 1'b1;
end

```

```

end
GAME_PROCESS_MOVE_INIT: begin
    sig_resetIteration      = 1'b1;
    Id_iterationCounter      = 1'b1;
end
GAME_PROCESS_ITER_CHECK: begin
    sig_iterationCheck      = 1'b1;
end
GAME_PROCESS_SET_POS: begin
    sig_setCurrentPOS       = 1'b1;
    Id_gameBoard_cur_X      = 1'b1;
    Id_gameBoard_cur_Y      = 1'b1;
    Id_gameBoard_cur_Value  = 1'b1;
end
GAME_SET_POS_WAIT: begin
    sig_IdExt                = 1'b1;
    Id_gameBoard_cur_Value  = 1'b1;
end
GAME_SET_POS_WAIT_2: begin
    sig_IdExt                = 1'b1;
    Id_gameBoard_cur_Value  = 1'b1;
    sig_getHighscore        = 1'b1;
    Id_highscore            = 1'b1;
end
GAME_PROCESS_WITHIN_BOUND: begin
    sig_checkBound          = 1'b1;
end
GAME_PROCESS_SET_NEXT_POS: begin

```

```

        sig_setCurrentNextPOS    = 1'b1;
        ld_gameBoard_next_X      = 1'b1;
        ld_gameBoard_next_Y      = 1'b1;
    end
    GAME_PROCESS_CALC_PRE: begin
        sig_calcMove              = 1'b1;
    end
    GAME_PROCESS_CALC: begin
        sig_calcMove              = 1'b1;
    end
    GAME_PROCESS_NO_MOVE: begin
        sig_noMove                 = 1'b1;
        ld_gameBoard_cur_X         = 1'b1;
        ld_gameBoard_cur_Y         = 1'b1;
        ld_gameBoard_cur_Value     = 1'b1;
        sig_getHighscore           = 1'b1;
        ld_highscore               = 1'b1;
    end
    NO_MOVE_LD_EXT: begin
        sig_ldExt                  = 1'b1;
        ld_gameBoard_cur_Value     = 1'b1;
        sig_getHighscore           = 1'b1;
        ld_highscore               = 1'b1;
    end
    NO_MOVE_LD_EXT_2: begin
        sig_ldExt                  = 1'b1;
        ld_gameBoard_cur_Value     = 1'b1;
        sig_getHighscore           = 1'b1;
    end

```



```

        Id_highscore                = 1'b1;
    end
//    DEBUG_WAIT_NO_MOVE: begin
//    end
    GAME_PROCESS_MERGE_MOVE_UPDATE_NEXT: begin
        sig_mergeUpdateNext        = 1'b1;
        gameRAM_writeEn            = 1'b1;
    end
    GAME_PROCESS_MERGE_MOVE_UPDATE_NEXT_EXT: begin
        sig_mergeUpdateNext        = 1'b1;
        gameRAM_writeEn            = 1'b1;
    end
    GAME_PROCESS_MERGE_MOVE_UPDATE_CUR: begin
        sig_mergeUpdateCur        = 1'b1;
        gameRAM_writeEn            = 1'b1;
    end
    GAME_PROCESS_MERGE_MOVE_UPDATE_CUR_EXT: begin
        sig_mergeUpdateCur        = 1'b1;
        gameRAM_writeEn            = 1'b1;
    end
    GAME_PROCESS_JUST_MOVE_UPDATE_NEXT: begin
        sig_noMergeUpdateNext     = 1'b1;
        gameRAM_writeEn            = 1'b1;
    end
    GAME_PROCESS_JUST_MOVE_UPDATE_NEXT_EXT: begin
        sig_noMergeUpdateNext     = 1'b1;
        gameRAM_writeEn            = 1'b1;
    end
end

```

```

GAME_PROCESS_JUST_MOVE_UPDATE_CUR: begin
    sig_noMergeUpdateCur        = 1'b1;
    gameRAM_writeEn              = 1'b1;
end

GAME_PROCESS_JUST_MOVE_UPDATE_CUR_EXT: begin
    sig_noMergeUpdateCur        = 1'b1;
    gameRAM_writeEn              = 1'b1;
end

GAME_PROCESS_ITER_INCRE: begin
    sig_iterationIncre           = 1'b1;
    Id_iterationCounter          = 1'b1;
end

GAME_RAND_NUM: begin
    LFBSR_enable                 = 1'b1;
    Id_randomNum                 = 1'b1;
end

GAME_CHECK_NUM: begin
    sig_checkRandNum             = 1'b1;
end

GAME_CHECK_NUM_WAIT: begin
    sig_checkRandNum             = 1'b1;
end

GAME_SPAWN_NUM: begin
    sig_spawnNumOnBoard          = 1'b1;
    gameRAM_writeEn              = 1'b1;
end

DEBUG_DISPLAY_BOARD: begin
    sig_debug_displayBoard       = 1'b1;
end

```

```

end
DEBUG_DISPLAY_BOARD_2: begin
    sig_debug_displayBoard = 1'b1;
end
GAME_DRAW_INIT: begin
    sig_drawBoard_init      = 1'b1;
end
GAME_DRAW_COUNTER_CHECK: begin
    sig_drawBoard_CounterCheck = 1'b1;
end
GAME_DRAW_LD_XY: begin
    sig_getCur_XY           = 1'b1;
    ld_gameBoard_cur_X       = 1'b1;
    ld_gameBoard_cur_Y       = 1'b1;
end
GAME_DRAW_LD_VAL: begin
    sig_ldExt                 = 1'b1;
    ld_gameBoard_cur_Value = 1'b1;
end
GAME_DRAW_LD_VAL_2: begin
    sig_ldExt                 = 1'b1;
    ld_gameBoard_cur_Value = 1'b1;
end
GAME_DRAW: begin
    sig_drawBoard             = 1'b1;
    writeEn                   = 1'b1;
end
GAME_DRAW_COUNTER_INCRE: begin

```

```

        sig_drawBoard_CounterEn = 1'b1;
    end
endcase
end

always@(posedge CLOCK_50)
begin: state_FF
    if(!resetn)
        current_state <= INIT_CLEAR_BOARD;
    else
        current_state <= next_state;
    end // state_FF

    assign stateLEDs = {{3'b0}, current_state};

endmodule

module datapath(
    // Standard I/O
    input CLOCK_50,
    input resetn,

    input [3:0] sig_move,

    // Signals from control
    input sig_clearBoard,
    input sig_checkRandNum,
    input sig_spawnNumOnBoard,

```

```

input sig_drawBoard,
input sig_drawBoard_init,
input sig_initDraw,
input sig_gameDraw,
input sig_gameEndDraw,
input sig_resetIteration,
input sig_iterationCheck,
input sig_setCurrentPOS,
input sig_setCurrentNextPOS,
input sig_checkBound,
input sig_calcMove,
input sig_noMove,
input sig_IdExt,////////////////////////////////////
input sig_mergeUpdateNext,
input sig_mergeUpdateCur,
input sig_noMergeUpdateNext,
input sig_noMergeUpdateCur,
input sig_iterationIncre,
input sig_debug_displayBoard,////////////////////////////////////
input sig_drawBoard_CounterCheck,
input sig_getCur_XY,
input sig_drawBoard_CounterEn,
input gameRAM_writeEn,
input Id_randomNum,
input Id_move,
input Id_iterationCounter,
input Id_gameBoard_cur_X,
input Id_gameBoard_cur_Y,

```

```

input Id_gameBoard_cur_Value,
input Id_gameBoard_next_X,
input Id_gameBoard_next_Y,
input LFBSR_enable,

// Signals to control
output reg sig_clearBoard_DONE,
output reg sig_randNum_GOOD,
output reg sig_drawBoard_DONE,
output reg sig_doneProcess,
output reg sig_toNoMove,
output reg sig_toMergeMove,
output reg sig_toJustMove,
output reg sig_nextIteration,
output reg sig_debug_displayBoard_DONE,
////////////////////////////////////
output reg sig_drawBoard_Cont,

// Game RAM I/O
input [11:0] gameRAM_DataOut,
output reg [11:0] gameRAM_DataIn,
output reg [3:0] gameRAM_Addr,

// VGA output
output reg [8:0] x,
output reg [7:0] y,
output reg [2:0] colour,
// output reg [8:0] screen_X,
// output reg [7:0] screen_Y,

```

```
// output reg [2:0] pixel_colour,
```

```
output reg [5:0] effective_X,
```

```
output reg [5:0] effective_Y,
```

```
output reg [5:0] rand_eff_X,
```

```
output reg [5:0] rand_eff_Y,
```

```
//////////
```

Debug signals

```
output reg [1:0] gameBoard_cur_X, gameBoard_next_X, gameBoard_cur_Y,  
gameBoard_next_Y, temp_X, temp_Y,
```

```
output reg [11:0] gameBoard_cur_Value,
```

```
output reg [3:0] randomNum_reg,
```

```
output [4:0] randomNum,
```

```
output reg [6:0] casc_Counter, temp_casc_Counter,
```

```
output reg sig_doneCasc,
```

```
output reg sig_randNumDraw_DONE,
```

```
input sig_cascCounter_init,
```

```
input ld_cascCounter,
```

```
input sig_casc_CounterCheck,
```

```
input sig_pixelCounter_init,
```

```
input sig_pixel_CounterEn,
```

```
input sig_drawRandNum,
```

```

input sel_randNum_XY,
input sel_randNum_Colour,
input sig_cascCounter_Incre,

input sig_getHighscore,
input ld_highscore,

output reg sig_gameLose,
// output reg [3:0] badMoves

input sig_drawEnd,

input colourful
);

// reg [3:0] randomNum_reg;
reg [3:0] move_reg;
// reg [1:0] gameBoard_cur_X;
// reg [1:0] gameBoard_next_X;
// reg [1:0] temp_X;
// reg [1:0] gameBoard_cur_Y;
// reg [1:0] gameBoard_next_Y;
// reg [1:0] temp_Y;
// reg [11:0] gameBoard_cur_Value;

// wire [3:0] randomNum;

// To VGA

```



```

    reg [8:0] screen_X;
    reg [7:0] screen_Y;
    reg [2:0] pixel_colour;
//    reg [5:0] effective_X;
//    reg [5:0] effective_Y;
    reg [8:0] rand_X;
    reg [7:0] rand_Y;
    reg [2:0] rand_colour;

    reg [11:0] highscore, temp_highscore;

    wire [11:0] randNum_12b;

    Linear_FB_Shift_Reg_12b randColour(
        .CLOCK_50(CLOCK_50),
        .resetn(resetn),
        .LFBSR_enable(LFBSR_enable),
        .out(randNum_12b)
    );

//    reg [5:0] rand_eff_X;
//    reg [5:0] rand_eff_Y;

    always @ (*) begin
        if (sel_randNum_XY) begin
            x = rand_X;
            y = rand_Y;
        end
    end

```

```

        else begin
            x = screen_X;
            y = screen_Y;
        end
        if (sel_randNum_Colour) begin
            colour = rand_colour;
        end
        else begin
            colour = pixel_colour;
        end
    end

// Iteration counter, loops through number of times we looked at the board
reg [2:0] iteration_Counter, temp_iter_counter;
// reg [6:0] casc_Counter, temp_casc_Counter;

// Registers
always@(posedge CLOCK_50) begin
    if(!resetn) begin
        highscore <= 12'b0;
        randomNum_reg <= 4'b0;
        move_reg <= 4'b0;
        iteration_Counter <= 3'b0;
        casc_Counter <= 7'b0;
    end
    else begin
        if(Id_highscore) begin
            highscore <= temp_highscore;
        end
    end
end

```

```

end
if(ld_randomNum) begin
    randomNum_reg <= randomNum[4:1];
end
if(ld_move) begin
    move_reg <= sig_move;
end
if(ld_iterationCounter) begin
    iteration_Counter <= temp_iter_counter;
end
if(ld_gameBoard_cur_X) begin
    gameBoard_cur_X <= temp_X;
end
if(ld_gameBoard_cur_Y) begin
    gameBoard_cur_Y <= temp_Y;
end
if(ld_gameBoard_cur_Value) begin
    gameBoard_cur_Value <= gameRAM_DataOut;
end
if(ld_gameBoard_next_X) begin
    gameBoard_next_X <= temp_X;
end
if(ld_gameBoard_next_Y) begin
    gameBoard_next_Y <= temp_Y;
end
if(ld_cascCounter) begin
    casc_Counter <= temp_casc_Counter;
end

```

```

        end
    end

    // INIT state
    // Clearing board
    reg [4:0] clearBoard_Counter;

    always @(posedge CLOCK_50) begin
        if (!resetn) begin
            clearBoard_Counter <= 5'b0;
        end
        else if (clearBoard_Counter == 5'b10000) begin
            clearBoard_Counter <= 5'b0;
        end
        else if (sig_clearBoard) begin
            clearBoard_Counter <= clearBoard_Counter + 1'b1;
        end
    end

    /////////////////////////////////////////////////// Debug display board
    reg [4:0] displayBoard_Counter;

    always @(posedge CLOCK_50) begin
        if (!resetn) begin
            displayBoard_Counter <= 5'b0;
        end
        else if (displayBoard_Counter == 5'b10000) begin
            displayBoard_Counter <= 5'b0;
        end
    end

```

```

        end
        else if (sig_debug_displayBoard) begin
            displayBoard_Counter <= displayBoard_Counter + 1'b1;
        end
    end

always @(posedge CLOCK_50, negedge resetn) begin
    if (!resetn) begin
        rand_X <= 9'd0;
        rand_Y <= 8'd0;
    end
    else if (sig_pixelCounter_init) begin
        rand_X <= ((randomNum_reg[3:2] * 6'd59) + 2'd3);
        rand_Y <= ((randomNum_reg[1:0] * 6'd59) + 2'd3);
    end
    else if (sig_pixel_CounterEn) begin
        if (((rand_X - 2'd3) - (((randomNum_reg[3:2]) * (6'd59)))) ==
casc_Counter) begin
            rand_X <= ((randomNum_reg[3:2] * 6'd59) + 2'd3);
            rand_Y <= rand_Y + 1'b1;
        end
        else begin
            rand_X <= rand_X + 1'b1;
        end
    end
end

// Random number gen

```

```

        Linear_FB_Shift_Reg_5b
randGen(////////////////////////////////////
////

        .CLOCK_50(CLOCK_50),
        .resetn(resetn),
        .LFBSR_enable(LFBSR_enable),
        .out(randomNum)

    );

//    counter_4b fakeRandGen(
//        .CLOCK_50(CLOCK_50),
//        .resetn(resetn),
//        .counter_4b_enable(LFBSR_enable),
//        .out(randomNum[4:1])
//    );

// Implement if we want to generate 2 new random numbers each time
//    reg [1:0] initRand_Counter;
//
//    always @(posedge CLOCK_50) begin
//        if (!resetn) begin
//            initRand_Counter <= 2'b0;
//        end
//        else if (initRand_Counter == 2'b10) begin
//            initRand_Counter <= 2'b0;
//        end
//        else if () begin
//            initRand_Counter <= initRand_Counter + 1'b1;
//        end

```

```

//      end

// Counter to track number of random numbers generated
reg [6:0] randNum_counter;

always @(posedge CLOCK_50) begin
    if (!resetn) begin
        randNum_counter <= 7'b0;
    end
    else if (sig_spawnNumOnBoard) begin
        randNum_counter <= 7'b0;
    end
    else if (LFBSR_enable) begin
        randNum_counter <= randNum_counter + 1'b1;
    end
end

end

// All calculations / checking
always @(*)
begin : ALU
    sig_randNum_GOOD = 1'b0;
    sig_doneProcess = 1'b0;
    sig_toNoMove = 1'b0;
    sig_toMergeMove = 1'b0;
    sig_toJustMove = 1'b0;
    sig_nextIteration = 1'b0;
    temp_X = 2'b0;
    temp_Y = 2'b0;

```

```

temp_iter_counter = 3'b0;
sig_drawBoard_Cont = 1'b1;
sig_doneCasc = 1'b0;
temp_casc_Counter = 7'b0;
temp_highscore = 12'b0;
sig_gameLose = 1'b0;
if (sig_getHighscore) begin
    if (gameBoard_cur_Value > highscore) begin
        temp_highscore = gameBoard_cur_Value;
    end
    else if (sig_clearBoard) begin
        temp_highscore = 12'b0;
    end
    else begin
        temp_highscore = highscore;
    end
end
if (sig_checkRandNum) begin
    if (gameRAM_DataOut == 12'd0) begin
        sig_randNum_GOOD = 1'b1;
    end
    if (randNum_counter == 7'd100) begin
        sig_gameLose = 1'b1;
    end
end
if (sig_resetIteration) begin
    temp_iter_counter = 3'b0;
end

```



```

        if (sig_iterationCheck) begin
            if (iteration_Counter == 3'b101) begin
                sig_doneProcess = 1'b1;
            end
        end
    end
    if (sig_setCurrentPOS) begin
        case (move_reg)
            4'b1000: ; // Left: x: 00 -> 11,
starting y don't care
            4'b0100: ; // Up: starting x
don't care, y: 00 -> 11
            4'b0010: temp_Y = 2'b11; // Down: starting x don't
care, y: 11 -> 00
            4'b0001: temp_X = 2'b11; // Right: x: 11 -> 00, starting y
don't care

        endcase
        // temp_X =
        2'b00;////////////////////////////////////
        // temp_Y =
        2'b11;////////////////////////////////////
    end
    if (sig_checkBound) begin
        if ((gameBoard_cur_X == 2'b00 && move_reg[3]) || // Left:
check from left to right, left most row no need to check
            (gameBoard_cur_Y == 2'b00 && move_reg[2]) ||
// Up: check from up to down, top most row no need to check
            (gameBoard_cur_Y == 2'b11 && move_reg[1]) ||
// Down: check from down to up, bottom most row no need to check
            (gameBoard_cur_X == 2'b11 && move_reg[0])) begin//
Right: check from right to left, right most row no need to check
            sig_toNoMove = 1'b1;
        end
    end

```

```

end
if (sig_setCurrentNextPOS) begin
    temp_X = gameBoard_cur_X;
    temp_Y = gameBoard_cur_Y;
    case (move_reg)
        4'b1000: temp_X = gameBoard_cur_X - 1'b1; // Left: Look
to -x
        4'b0100: temp_Y = gameBoard_cur_Y - 1'b1; // Up: Look to
-y
        4'b0010: temp_Y = gameBoard_cur_Y + 1'b1; //
Down: Look to +y
        4'b0001: temp_X = gameBoard_cur_X + 1'b1; // Right: Look
to +x
    endcase
end
if (sig_calcMove) begin
    if (gameRAM_DataOut == 12'b0) begin
        sig_toJustMove = 1'b1;
    end
    else if (gameRAM_DataOut == gameBoard_cur_Value) begin
        sig_toMergeMove = 1'b1;
    end
    else begin
        sig_toNoMove = 1'b1;
    end
end
if (sig_noMove) begin
    case (move_reg)
        4'b1000: begin // Left: Cycles through y, move right one
row in y when done

```

```

== 2'b11) begin
    if (gameBoard_cur_X == 2'b11 && gameBoard_cur_Y
        sig_nextIteration = 1'b1;
    end
    else if (gameBoard_cur_Y == 2'b11) begin
        temp_Y = 2'b00;
        temp_X = gameBoard_cur_X + 1'b1;
    end
    else begin
        temp_X = gameBoard_cur_X;
        temp_Y = gameBoard_cur_Y + 1'b1;
    end
end
4'b0100: begin // Up: Cycles through x, move down one row
    if (gameBoard_cur_X == 2'b11 && gameBoard_cur_Y
        sig_nextIteration = 1'b1;
    end
    else if (gameBoard_cur_X == 2'b11) begin
        temp_X = 2'b00;
        temp_Y = gameBoard_cur_Y + 1'b1;
    end
    else begin
        temp_Y = gameBoard_cur_Y;
        temp_X = gameBoard_cur_X + 1'b1;
    end
end
4'b0010: begin // Down: Cycles through x, move up one
row in y when done

```

```

== 2'b00) begin
    if (gameBoard_cur_X == 2'b11 && gameBoard_cur_Y
        sig_nextIteration = 1'b1;
    end
    else if (gameBoard_cur_X == 2'b11) begin
        temp_X = 2'b00;
        temp_Y = gameBoard_cur_Y - 1'b1;
    end
    else begin
        temp_Y = gameBoard_cur_Y;
        temp_X = gameBoard_cur_X + 1'b1;
    end
end
end
4'b0001: begin    // Right: Cycles through y, move left one
    if (gameBoard_cur_X == 2'b00 && gameBoard_cur_Y
        sig_nextIteration = 1'b1;
    end
    else if (gameBoard_cur_Y == 2'b11) begin
        temp_Y = 2'b00;
        temp_X = gameBoard_cur_X - 1'b1;
    end
    else begin
        temp_X = gameBoard_cur_X;
        temp_Y = gameBoard_cur_Y + 1'b1;
    end
end
end
endcase

```

row in y when done

== 2'b11) begin

```

end
if (sig_iterationIncre) begin
    temp_iter_counter = iteration_Counter + 1'b1;
end
if (sig_getCur_XY) begin
    if (screen_X >= 9'd3 && screen_X <= 9'd59) begin
        temp_X = 2'b00;
    end
    else if (screen_X >= 9'd62 && screen_X <= 9'd118) begin
        temp_X = 2'b01;
    end
    else if (screen_X >= 9'd121 && screen_X <= 9'd177) begin
        temp_X = 2'b10;
    end
    else if (screen_X >= 9'd180 && screen_X <= 9'd236) begin
        temp_X = 2'b11;
    end
    if (screen_Y >= 8'd3 && screen_Y <= 8'd59) begin
        temp_Y = 2'b00;
    end
    else if (screen_Y >= 8'd62 && screen_Y <= 8'd118) begin
        temp_Y = 2'b01;
    end
    else if (screen_Y >= 8'd121 && screen_Y <= 8'd177) begin
        temp_Y = 2'b10;
    end
    else if (screen_Y >= 8'd180 && screen_Y <= 8'd236) begin
        temp_Y = 2'b11;
    end
end

```

```

        end
    end
    if (sig_drawBoard_CounterCheck) begin
        if (screen_Y == 8'd240) begin
            sig_drawBoard_Cont = 1'b0;
        end
    end
    if (sig_cascCounter_init) begin
        temp_casc_Counter = 7'b0;
    end
    if (sig_casc_CounterCheck) begin
        if (casc_Counter == 7'd57) begin
            sig_doneCasc = 1'b1;
        end
    end
    if (sig_cascCounter_Incre) begin
        temp_casc_Counter = casc_Counter + 1'b1;
    end
end

// Game RAM address mux
always @(*) begin
    gameRAM_Addr = 4'b0;
    if (sig_clearBoard) begin
        gameRAM_Addr = clearBoard_Counter[3:0];
    end
    if (sig_checkRandNum) begin
        gameRAM_Addr = randomNum_reg;
    end
end

```

```

end
if (sig_spawnNumOnBoard) begin
    gameRAM_Addr = randomNum_reg;
end
if (sig_setCurrentPOS) begin
    gameRAM_Addr = {gameBoard_cur_X, gameBoard_cur_Y};
end
if (sig_calcMove) begin
    gameRAM_Addr = {gameBoard_next_X, gameBoard_next_Y};
end
if (sig_IdExt) begin
    gameRAM_Addr = {gameBoard_cur_X, gameBoard_cur_Y};
end
if (sig_mergeUpdateNext) begin
    gameRAM_Addr = {gameBoard_next_X, gameBoard_next_Y};
end
if (sig_mergeUpdateCur) begin
    gameRAM_Addr = {gameBoard_cur_X, gameBoard_cur_Y};
end
if (sig_noMergeUpdateNext) begin
    gameRAM_Addr = {gameBoard_next_X, gameBoard_next_Y};
end
if (sig_noMergeUpdateCur) begin
    gameRAM_Addr = {gameBoard_cur_X, gameBoard_cur_Y};
end
if (sig_drawBoard) begin
    gameRAM_Addr = {gameBoard_cur_X, gameBoard_cur_Y};
end

```

```

        if (sig_debug_displayBoard) begin////////////////////////////////////////
            gameRAM_Addr = displayBoard_Counter[3:0];
        end
//        if (sig_noMove) begin
//            gameRAM_Addr = {gameBoard_cur_X, gameBoard_cur_Y};
//        end
    end

// Game RAM DataIn mux
always @(*) begin
    gameRAM_DataIn = 12'd0;
    if (sig_clearBoard) begin
        gameRAM_DataIn = 12'd0;
    end
    if (sig_spawnNumOnBoard) begin
        gameRAM_DataIn = 12'd2;
    end
    if (sig_mergeUpdateNext) begin
        gameRAM_DataIn = gameBoard_cur_Value << 1; // Note 4096 +
4096 = 0 due to overflow
    end
    if (sig_mergeUpdateCur) begin
        gameRAM_DataIn = 12'd0;
    end
    if (sig_noMergeUpdateNext) begin
        gameRAM_DataIn = gameBoard_cur_Value;
    end
    if (sig_noMergeUpdateCur) begin
        gameRAM_DataIn = 12'd0;
    end
end

```



```

        end

    end

    // Done signal outputs
    always @(*) begin
        if (!resetn) begin
            sig_clearBoard_DONE = 1'b0;
            sig_drawBoard_DONE = 1'b0;
            sig_debug_displayBoard_DONE = 1'b0;
            sig_randNumDraw_DONE = 1'b0;
        end
        else begin
            sig_clearBoard_DONE = (clearBoard_Counter == 5'b10000);
            sig_drawBoard_DONE = (screen_Y == 8'd240);
            sig_debug_displayBoard_DONE = (displayBoard_Counter ==
5'b10000);
            sig_randNumDraw_DONE = (((rand_X - 2'd3) -
((randomNum_reg[3:2] * (6'd59))) >= casc_Counter) && (((rand_Y - 2'd3) -
((randomNum_reg[1:0] * (6'd59))) >= casc_Counter));
        end
    end

    // Iterating through the game board area
    always @(posedge CLOCK_50, negedge resetn) begin
        if (!resetn) begin
            screen_X <= 9'd0;
            screen_Y <= 8'd0;
        end
        else if (sig_drawBoard_init) begin

```

```

        screen_X <= 9'd0;
        screen_Y <= 8'd0;
    end
    else if (sig_drawBoard_CounterEn) begin
        if(screen_X == 9'd318) begin
            screen_X <= 9'd0;
            screen_Y <= screen_Y + 1'b1;
        end
        else begin
            screen_X <= screen_X + 1'b1;
        end
    end
end

always @(*) begin
    pixel_colour = 3'b000;
    rand_colour = 3'b000;
    if (sig_drawEnd) begin
        if (screen_X >= 9'd240 && screen_X <= 9'd319 && screen_Y <=
8'd119) begin
            if ((screen_X == 9'd247 && screen_Y == 8'd8) ||
                (screen_X == 9'd247 && screen_Y == 8'd9) ||
                (screen_X == 9'd247 && screen_Y == 8'd10) ||
                (screen_X == 9'd247 && screen_Y == 8'd11) ||
                (screen_X == 9'd247 && screen_Y == 8'd12) ||
                (screen_X == 9'd247 && screen_Y == 8'd13) ||
                (screen_X == 9'd247 && screen_Y == 8'd14) ||
                (screen_X == 9'd247 && screen_Y == 8'd15) ||
                (screen_X == 9'd247 && screen_Y == 8'd16) ||

```

```

(screen_X == 9'd247 && screen_Y == 8'd17) ||
(screen_X == 9'd247 && screen_Y == 8'd18) ||
(screen_X == 9'd247 && screen_Y == 8'd19) ||
(screen_X == 9'd247 && screen_Y == 8'd20) ||
(screen_X == 9'd248 && screen_Y == 8'd13) ||
(screen_X == 9'd249 && screen_Y == 8'd13) ||
(screen_X == 9'd250 && screen_Y == 8'd13) ||
(screen_X == 9'd251 && screen_Y == 8'd13) ||
(screen_X == 9'd252 && screen_Y == 8'd13) ||
(screen_X == 9'd253 && screen_Y == 8'd13) ||
(screen_X == 9'd254 && screen_Y == 8'd13) ||
(screen_X == 9'd255 && screen_Y == 8'd8) ||
(screen_X == 9'd255 && screen_Y == 8'd9) ||
(screen_X == 9'd255 && screen_Y == 8'd10) ||
(screen_X == 9'd255 && screen_Y == 8'd11) ||
(screen_X == 9'd255 && screen_Y == 8'd12) ||
(screen_X == 9'd255 && screen_Y == 8'd13) ||
(screen_X == 9'd255 && screen_Y == 8'd14) ||
(screen_X == 9'd255 && screen_Y == 8'd15) ||
(screen_X == 9'd255 && screen_Y == 8'd16) ||
(screen_X == 9'd255 && screen_Y == 8'd17) ||
(screen_X == 9'd255 && screen_Y == 8'd18) ||
(screen_X == 9'd255 && screen_Y == 8'd19) ||
(screen_X == 9'd255 && screen_Y == 8'd20)
) begin // H
    pixel_colour = colourful ? ((randNum_12b[11:9] ==
3'b0) ? 3'b111 : randNum_12b[11:9]) : 3'b111;
end
if ((screen_X == 9'd258 && screen_Y == 8'd9) ||

```

```

(screen_X == 9'd258 && screen_Y == 8'd12) ||
(screen_X == 9'd258 && screen_Y == 8'd13) ||
(screen_X == 9'd258 && screen_Y == 8'd14) ||
(screen_X == 9'd258 && screen_Y == 8'd15) ||
(screen_X == 9'd258 && screen_Y == 8'd16) ||
(screen_X == 9'd258 && screen_Y == 8'd17) ||
(screen_X == 9'd258 && screen_Y == 8'd18)
) begin // i
    pixel_colour = colourful ? ((randNum_12b[10:8] ==
3'b0) ? 3'b111 : randNum_12b[10:8]) : 3'b111;
end
if ((screen_X == 9'd261 && screen_Y == 8'd12) ||
    (screen_X == 9'd261 && screen_Y == 8'd13) ||
    (screen_X == 9'd261 && screen_Y == 8'd14) ||
    (screen_X == 9'd261 && screen_Y == 8'd15) ||
    (screen_X == 9'd261 && screen_Y == 8'd16) ||
    (screen_X == 9'd261 && screen_Y == 8'd17) ||
    (screen_X == 9'd261 && screen_Y == 8'd18) ||
    (screen_X == 9'd265 && screen_Y == 8'd12) ||
    (screen_X == 9'd265 && screen_Y == 8'd13) ||
    (screen_X == 9'd265 && screen_Y == 8'd14) ||
    (screen_X == 9'd265 && screen_Y == 8'd15) ||
    (screen_X == 9'd265 && screen_Y == 8'd16) ||
    (screen_X == 9'd265 && screen_Y == 8'd17) ||
    (screen_X == 9'd265 && screen_Y == 8'd18) ||
    (screen_X == 9'd265 && screen_Y == 8'd19) ||
    (screen_X == 9'd265 && screen_Y == 8'd20) ||
    (screen_X == 9'd265 && screen_Y == 8'd21) ||
    (screen_X == 9'd265 && screen_Y == 8'd22) ||

```

```

(screen_X == 9'd265 && screen_Y == 8'd23) ||
(screen_X == 9'd265 && screen_Y == 8'd24) ||
(screen_X == 9'd262 && screen_Y == 8'd12) ||
(screen_X == 9'd263 && screen_Y == 8'd12) ||
(screen_X == 9'd264 && screen_Y == 8'd12) ||
(screen_X == 9'd262 && screen_Y == 8'd18) ||
(screen_X == 9'd263 && screen_Y == 8'd18) ||
(screen_X == 9'd264 && screen_Y == 8'd18) ||
(screen_X == 9'd261 && screen_Y == 8'd22) ||
(screen_X == 9'd261 && screen_Y == 8'd23) ||
(screen_X == 9'd261 && screen_Y == 8'd24) ||
(screen_X == 9'd262 && screen_Y == 8'd24) ||
(screen_X == 9'd263 && screen_Y == 8'd24) ||
(screen_X == 9'd264 && screen_Y == 8'd24)
) begin // g
    pixel_colour = colourful ? ((randNum_12b[9:7] ==
3'b0) ? 3'b111 : randNum_12b[9:7]) : 3'b111;
end
if ((screen_X == 9'd267 && screen_Y == 8'd8) ||
    (screen_X == 9'd268 && screen_Y == 8'd8) ||
    (screen_X == 9'd268 && screen_Y == 8'd9) ||
    (screen_X == 9'd268 && screen_Y == 8'd10) ||
    (screen_X == 9'd268 && screen_Y == 8'd11) ||
    (screen_X == 9'd268 && screen_Y == 8'd12) ||
    (screen_X == 9'd268 && screen_Y == 8'd13) ||
    (screen_X == 9'd268 && screen_Y == 8'd14) ||
    (screen_X == 9'd268 && screen_Y == 8'd15) ||
    (screen_X == 9'd268 && screen_Y == 8'd16) ||
    (screen_X == 9'd268 && screen_Y == 8'd17) ||

```

```

(screen_X == 9'd268 && screen_Y == 8'd18) ||
(screen_X == 9'd268 && screen_Y == 8'd19) ||
(screen_X == 9'd269 && screen_Y == 8'd13) ||
(screen_X == 9'd270 && screen_Y == 8'd13) ||
(screen_X == 9'd271 && screen_Y == 8'd13) ||
(screen_X == 9'd272 && screen_Y == 8'd13) ||
(screen_X == 9'd272 && screen_Y == 8'd14) ||
(screen_X == 9'd272 && screen_Y == 8'd15) ||
(screen_X == 9'd272 && screen_Y == 8'd16) ||
(screen_X == 9'd272 && screen_Y == 8'd17) ||
(screen_X == 9'd272 && screen_Y == 8'd18)
) begin // h
    pixel_colour = colourful ? ((randNum_12b[8:6] ==
3'b0) ? 3'b111 : randNum_12b[8:6]) : 3'b111;
end
if ((screen_X == 9'd276 && screen_Y == 8'd12) ||
(screen_X == 9'd277 && screen_Y == 8'd12) ||
(screen_X == 9'd278 && screen_Y == 8'd12) ||
(screen_X == 9'd279 && screen_Y == 8'd12) ||
(screen_X == 9'd280 && screen_Y == 8'd12) ||
(screen_X == 9'd281 && screen_Y == 8'd12) ||
(screen_X == 9'd276 && screen_Y == 8'd13) ||
(screen_X == 9'd276 && screen_Y == 8'd14) ||
(screen_X == 9'd276 && screen_Y == 8'd15) ||
(screen_X == 9'd276 && screen_Y == 8'd16) ||
(screen_X == 9'd276 && screen_Y == 8'd17) ||
(screen_X == 9'd276 && screen_Y == 8'd18) ||
(screen_X == 9'd276 && screen_Y == 8'd19) ||
(screen_X == 9'd277 && screen_Y == 8'd15) ||

```

```

(screen_X == 9'd278 && screen_Y == 8'd15) ||
(screen_X == 9'd279 && screen_Y == 8'd15) ||
(screen_X == 9'd280 && screen_Y == 8'd15) ||
(screen_X == 9'd281 && screen_Y == 8'd15) ||
(screen_X == 9'd282 && screen_Y == 8'd15) ||
(screen_X == 9'd277 && screen_Y == 8'd19) ||
(screen_X == 9'd278 && screen_Y == 8'd19) ||
(screen_X == 9'd279 && screen_Y == 8'd19) ||
(screen_X == 9'd280 && screen_Y == 8'd19) ||
(screen_X == 9'd281 && screen_Y == 8'd19) ||
(screen_X == 9'd282 && screen_Y == 8'd19) ||
(screen_X == 9'd281 && screen_Y == 8'd13) ||
(screen_X == 9'd282 && screen_Y == 8'd13) ||
(screen_X == 9'd282 && screen_Y == 8'd14)
) begin // e
    pixel_colour = colourful ? ((randNum_12b[7:5] ==
3'b0) ? 3'b111 : randNum_12b[7:5]) : 3'b111;
end
if ((screen_X == 9'd286 && screen_Y == 8'd13) ||
    (screen_X == 9'd287 && screen_Y == 8'd13) ||
    (screen_X == 9'd288 && screen_Y == 8'd13) ||
    (screen_X == 9'd289 && screen_Y == 8'd13) ||
    (screen_X == 9'd290 && screen_Y == 8'd13) ||
    (screen_X == 9'd291 && screen_Y == 8'd13) ||
    (screen_X == 9'd292 && screen_Y == 8'd13) ||
    (screen_X == 9'd293 && screen_Y == 8'd13) ||
    (screen_X == 9'd286 && screen_Y == 8'd14) ||
    (screen_X == 9'd286 && screen_Y == 8'd15) ||
    (screen_X == 9'd286 && screen_Y == 8'd16) ||

```

```

(screen_X == 9'd286 && screen_Y == 8'd17) ||
(screen_X == 9'd287 && screen_Y == 8'd17) ||
(screen_X == 9'd288 && screen_Y == 8'd17) ||
(screen_X == 9'd289 && screen_Y == 8'd17) ||
(screen_X == 9'd290 && screen_Y == 8'd17) ||
(screen_X == 9'd291 && screen_Y == 8'd17) ||
(screen_X == 9'd292 && screen_Y == 8'd17) ||
(screen_X == 9'd293 && screen_Y == 8'd17) ||
(screen_X == 9'd293 && screen_Y == 8'd18) ||
(screen_X == 9'd293 && screen_Y == 8'd19) ||
(screen_X == 9'd287 && screen_Y == 8'd20) ||
(screen_X == 9'd288 && screen_Y == 8'd20) ||
(screen_X == 9'd289 && screen_Y == 8'd20) ||
(screen_X == 9'd290 && screen_Y == 8'd20) ||
(screen_X == 9'd291 && screen_Y == 8'd20) ||
(screen_X == 9'd292 && screen_Y == 8'd20) ||
(screen_X == 9'd293 && screen_Y == 8'd20)
) begin // s
    pixel_colour = colourful ? ((randNum_12b[6:4] ==
3'b0) ? 3'b111 : randNum_12b[6:4]) : 3'b111;
end
if ((screen_X == 9'd295 && screen_Y == 8'd10) ||
    (screen_X == 9'd296 && screen_Y == 8'd10) ||
    (screen_X == 9'd297 && screen_Y == 8'd10) ||
    (screen_X == 9'd298 && screen_Y == 8'd10) ||
    (screen_X == 9'd299 && screen_Y == 8'd10) ||
    (screen_X == 9'd300 && screen_Y == 8'd10) ||
    (screen_X == 9'd301 && screen_Y == 8'd10) ||
    (screen_X == 9'd302 && screen_Y == 8'd10) ||

```



```

(screen_X == 9'd303 && screen_Y == 8'd10) ||
(screen_X == 9'd304 && screen_Y == 8'd10) ||
(screen_X == 9'd305 && screen_Y == 8'd10) ||
(screen_X == 9'd300 && screen_Y == 8'd11) ||
(screen_X == 9'd300 && screen_Y == 8'd12) ||
(screen_X == 9'd300 && screen_Y == 8'd13) ||
(screen_X == 9'd300 && screen_Y == 8'd14) ||
(screen_X == 9'd300 && screen_Y == 8'd15) ||
(screen_X == 9'd300 && screen_Y == 8'd16) ||
(screen_X == 9'd300 && screen_Y == 8'd17) ||
(screen_X == 9'd300 && screen_Y == 8'd18) ||
(screen_X == 9'd300 && screen_Y == 8'd19) ||
(screen_X == 9'd300 && screen_Y == 8'd20) ||
(screen_X == 9'd300 && screen_Y == 8'd21)
) begin // t
    pixel_colour = colourful ? ((randNum_12b[5:3] ==
3'b0) ? 3'b111 : randNum_12b[5:3]) : 3'b111;
end
if ((screen_X == 9'd258 && screen_Y == 8'd29) ||
    (screen_X == 9'd256 && screen_Y == 8'd30) ||
    (screen_X == 9'd257 && screen_Y == 8'd30) ||
    (screen_X == 9'd255 && screen_Y == 8'd31) ||
    (screen_X == 9'd255 && screen_Y == 8'd32) ||
    (screen_X == 9'd255 && screen_Y == 8'd33) ||
    (screen_X == 9'd255 && screen_Y == 8'd34) ||
    (screen_X == 9'd256 && screen_Y == 8'd35) ||
    (screen_X == 9'd256 && screen_Y == 8'd36) ||
    (screen_X == 9'd256 && screen_Y == 8'd37) ||
    (screen_X == 9'd257 && screen_Y == 8'd38) ||

```

```

(screen_X == 9'd257 && screen_Y == 8'd39) ||
(screen_X == 9'd258 && screen_Y == 8'd40) ||
(screen_X == 9'd258 && screen_Y == 8'd41) ||
(screen_X == 9'd259 && screen_Y == 8'd41) ||
(screen_X == 9'd259 && screen_Y == 8'd42) ||
(screen_X == 9'd260 && screen_Y == 8'd42) ||
(screen_X == 9'd254 && screen_Y == 8'd42) ||
(screen_X == 9'd255 && screen_Y == 8'd42) ||
(screen_X == 9'd256 && screen_Y == 8'd42) ||
(screen_X == 9'd257 && screen_Y == 8'd43) ||
(screen_X == 9'd258 && screen_Y == 8'd43) ||
(screen_X == 9'd259 && screen_Y == 8'd43) ||
(screen_X == 9'd260 && screen_Y == 8'd43) ||
(screen_X == 9'd261 && screen_Y == 8'd43) ||
(screen_X == 9'd261 && screen_Y == 8'd38) ||
(screen_X == 9'd261 && screen_Y == 8'd39) ||
(screen_X == 9'd261 && screen_Y == 8'd40) ||
(screen_X == 9'd261 && screen_Y == 8'd41) ||
(screen_X == 9'd261 && screen_Y == 8'd42)
) begin // arrow
    pixel_colour = colourful ? ((randNum_12b[4:3] ==
3'b0) ? 3'b111 : randNum_12b[4:3]) : 3'b111;
end
if ((screen_X == 9'd273 && screen_Y == 8'd29) ||
    (screen_X == 9'd274 && screen_Y == 8'd29) ||
    (screen_X == 9'd275 && screen_Y == 8'd29) ||
    (screen_X == 9'd276 && screen_Y == 8'd29) ||
    (screen_X == 9'd277 && screen_Y == 8'd29) ||
    (screen_X == 9'd278 && screen_Y == 8'd29) ||

```

(screen\_X == 9'd279 && screen\_Y == 8'd29) ||  
(screen\_X == 9'd280 && screen\_Y == 8'd29) ||  
(screen\_X == 9'd281 && screen\_Y == 8'd29) ||  
(screen\_X == 9'd281 && screen\_Y == 8'd30) ||  
(screen\_X == 9'd281 && screen\_Y == 8'd31) ||  
(screen\_X == 9'd281 && screen\_Y == 8'd36) ||  
(screen\_X == 9'd281 && screen\_Y == 8'd37) ||  
(screen\_X == 9'd281 && screen\_Y == 8'd38) ||  
(screen\_X == 9'd281 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd281 && screen\_Y == 8'd40) ||  
(screen\_X == 9'd281 && screen\_Y == 8'd41) ||  
(screen\_X == 9'd273 && screen\_Y == 8'd30) ||  
(screen\_X == 9'd273 && screen\_Y == 8'd31) ||  
(screen\_X == 9'd273 && screen\_Y == 8'd32) ||  
(screen\_X == 9'd273 && screen\_Y == 8'd33) ||  
(screen\_X == 9'd273 && screen\_Y == 8'd34) ||  
(screen\_X == 9'd273 && screen\_Y == 8'd35) ||  
(screen\_X == 9'd274 && screen\_Y == 8'd35) ||  
(screen\_X == 9'd275 && screen\_Y == 8'd35) ||  
(screen\_X == 9'd276 && screen\_Y == 8'd35) ||  
(screen\_X == 9'd277 && screen\_Y == 8'd35) ||  
(screen\_X == 9'd278 && screen\_Y == 8'd35) ||  
(screen\_X == 9'd279 && screen\_Y == 8'd35) ||  
(screen\_X == 9'd280 && screen\_Y == 8'd35) ||  
(screen\_X == 9'd281 && screen\_Y == 8'd35) ||  
(screen\_X == 9'd272 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd272 && screen\_Y == 8'd40) ||  
(screen\_X == 9'd272 && screen\_Y == 8'd41) ||

```

(screen_X == 9'd273 && screen_Y == 8'd41) ||
(screen_X == 9'd274 && screen_Y == 8'd41) ||
(screen_X == 9'd275 && screen_Y == 8'd41) ||
(screen_X == 9'd276 && screen_Y == 8'd41) ||
(screen_X == 9'd277 && screen_Y == 8'd41) ||
(screen_X == 9'd278 && screen_Y == 8'd41) ||
(screen_X == 9'd279 && screen_Y == 8'd41) ||
(screen_X == 9'd280 && screen_Y == 8'd41)
) begin // S
    pixel_colour = colourful ? ((randNum_12b[3:1] ==
3'b0) ? 3'b111 : randNum_12b[3:1]) : 3'b111;
end
if ((screen_X == 9'd283 && screen_Y == 8'd32) ||
    (screen_X == 9'd284 && screen_Y == 8'd32) ||
    (screen_X == 9'd285 && screen_Y == 8'd32) ||
    (screen_X == 9'd286 && screen_Y == 8'd32) ||
    (screen_X == 9'd287 && screen_Y == 8'd32) ||
    (screen_X == 9'd288 && screen_Y == 8'd32) ||
    (screen_X == 9'd288 && screen_Y == 8'd33) ||
    (screen_X == 9'd283 && screen_Y == 8'd33) ||
    (screen_X == 9'd283 && screen_Y == 8'd34) ||
    (screen_X == 9'd283 && screen_Y == 8'd35) ||
    (screen_X == 9'd283 && screen_Y == 8'd36) ||
    (screen_X == 9'd283 && screen_Y == 8'd37) ||
    (screen_X == 9'd283 && screen_Y == 8'd38) ||
    (screen_X == 9'd283 && screen_Y == 8'd39) ||
    (screen_X == 9'd284 && screen_Y == 8'd39) ||
    (screen_X == 9'd285 && screen_Y == 8'd39) ||
    (screen_X == 9'd286 && screen_Y == 8'd39) ||

```

```

(screen_X == 9'd287 && screen_Y == 8'd39) ||
(screen_X == 9'd288 && screen_Y == 8'd39)
) begin // c
    pixel_colour = colourful ? ((randNum_12b[2:0] ==
3'b0) ? 3'b111 : randNum_12b[2:0]) : 3'b111;
end
if ((screen_X == 9'd291 && screen_Y == 8'd32) ||
    (screen_X == 9'd292 && screen_Y == 8'd32) ||
    (screen_X == 9'd293 && screen_Y == 8'd32) ||
    (screen_X == 9'd294 && screen_Y == 8'd32) ||
    (screen_X == 9'd295 && screen_Y == 8'd32) ||
    (screen_X == 9'd296 && screen_Y == 8'd32) ||
    (screen_X == 9'd297 && screen_Y == 8'd32) ||
    (screen_X == 9'd292 && screen_Y == 8'd39) ||
    (screen_X == 9'd293 && screen_Y == 8'd39) ||
    (screen_X == 9'd294 && screen_Y == 8'd39) ||
    (screen_X == 9'd295 && screen_Y == 8'd39) ||
    (screen_X == 9'd296 && screen_Y == 8'd39) ||
    (screen_X == 9'd291 && screen_Y == 8'd33) ||
    (screen_X == 9'd291 && screen_Y == 8'd34) ||
    (screen_X == 9'd291 && screen_Y == 8'd35) ||
    (screen_X == 9'd291 && screen_Y == 8'd36) ||
    (screen_X == 9'd291 && screen_Y == 8'd37) ||
    (screen_X == 9'd291 && screen_Y == 8'd38) ||
    (screen_X == 9'd291 && screen_Y == 8'd39) ||
    (screen_X == 9'd297 && screen_Y == 8'd33) ||
    (screen_X == 9'd297 && screen_Y == 8'd34) ||
    (screen_X == 9'd297 && screen_Y == 8'd35) ||
    (screen_X == 9'd297 && screen_Y == 8'd36) ||

```

```

        (screen_X == 9'd297 && screen_Y == 8'd37) ||
        (screen_X == 9'd297 && screen_Y == 8'd38) ||
        (screen_X == 9'd297 && screen_Y == 8'd39) ||
        (screen_X == 9'd297 && screen_Y == 8'd40)
    ) begin // o
        pixel_colour = colourful ? (((randNum_12b[11:9] ^
randNum_12b[2:0]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^ randNum_12b[2:0])) :
3'b111;

    end

    if ((screen_X == 9'd300 && screen_Y == 8'd33) ||
        (screen_X == 9'd300 && screen_Y == 8'd34) ||
        (screen_X == 9'd300 && screen_Y == 8'd35) ||
        (screen_X == 9'd300 && screen_Y == 8'd36) ||
        (screen_X == 9'd300 && screen_Y == 8'd37) ||
        (screen_X == 9'd300 && screen_Y == 8'd38) ||
        (screen_X == 9'd300 && screen_Y == 8'd39) ||
        (screen_X == 9'd301 && screen_Y == 8'd34) ||
        (screen_X == 9'd302 && screen_Y == 8'd33) ||
        (screen_X == 9'd303 && screen_Y == 8'd33) ||
        (screen_X == 9'd304 && screen_Y == 8'd33) ||
        (screen_X == 9'd305 && screen_Y == 8'd33) ||
        (screen_X == 9'd306 && screen_Y == 8'd33)
    ) begin // r
        pixel_colour = colourful ? (((randNum_12b[10:8] ^
randNum_12b[2:0]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^ randNum_12b[2:0])) :
3'b111;

    end

    if ((screen_X == 9'd310 && screen_Y == 8'd32) ||
        (screen_X == 9'd311 && screen_Y == 8'd32) ||
        (screen_X == 9'd312 && screen_Y == 8'd32) ||

```

```

(screen_X == 9'd313 && screen_Y == 8'd32) ||
(screen_X == 9'd314 && screen_Y == 8'd33) ||
(screen_X == 9'd314 && screen_Y == 8'd34) ||
(screen_X == 9'd315 && screen_Y == 8'd34) ||
(screen_X == 9'd315 && screen_Y == 8'd35) ||
(screen_X == 9'd315 && screen_Y == 8'd36) ||
(screen_X == 9'd310 && screen_Y == 8'd37) ||
(screen_X == 9'd311 && screen_Y == 8'd37) ||
(screen_X == 9'd312 && screen_Y == 8'd37) ||
(screen_X == 9'd313 && screen_Y == 8'd37) ||
(screen_X == 9'd314 && screen_Y == 8'd37) ||
(screen_X == 9'd315 && screen_Y == 8'd37) ||
(screen_X == 9'd311 && screen_Y == 8'd41) ||
(screen_X == 9'd312 && screen_Y == 8'd41) ||
(screen_X == 9'd313 && screen_Y == 8'd41) ||
(screen_X == 9'd314 && screen_Y == 8'd41) ||
(screen_X == 9'd315 && screen_Y == 8'd41) ||
(screen_X == 9'd310 && screen_Y == 8'd40) ||
(screen_X == 9'd309 && screen_Y == 8'd33) ||
(screen_X == 9'd309 && screen_Y == 8'd34) ||
(screen_X == 9'd309 && screen_Y == 8'd35) ||
(screen_X == 9'd309 && screen_Y == 8'd36) ||
(screen_X == 9'd309 && screen_Y == 8'd37) ||
(screen_X == 9'd309 && screen_Y == 8'd38) ||
(screen_X == 9'd309 && screen_Y == 8'd39)
) begin // e

pixel_colour = colourful ? (((randNum_12b[11:9] ^
randNum_12b[3:1]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^ randNum_12b[3:1])) :
3'b111;

```

end

```
if ((screen_X == 9'd251 && screen_Y == 8'd50) ||
    (screen_X == 9'd252 && screen_Y == 8'd50) ||
    (screen_X == 9'd253 && screen_Y == 8'd50) ||
    (screen_X == 9'd254 && screen_Y == 8'd50) ||
    (screen_X == 9'd255 && screen_Y == 8'd50) ||
    (screen_X == 9'd256 && screen_Y == 8'd50) ||
    (screen_X == 9'd257 && screen_Y == 8'd50) ||
    (screen_X == 9'd258 && screen_Y == 8'd50) ||
    (screen_X == 9'd259 && screen_Y == 8'd50) ||
    (screen_X == 9'd260 && screen_Y == 8'd50) ||
    (screen_X == 9'd261 && screen_Y == 8'd50) ||
    (screen_X == 9'd262 && screen_Y == 8'd50) ||
    (screen_X == 9'd263 && screen_Y == 8'd50) ||
    (screen_X == 9'd264 && screen_Y == 8'd50) ||
    (screen_X == 9'd265 && screen_Y == 8'd50) ||
    (screen_X == 9'd266 && screen_Y == 8'd50) ||
    (screen_X == 9'd267 && screen_Y == 8'd50) ||
    (screen_X == 9'd268 && screen_Y == 8'd50) ||
    (screen_X == 9'd269 && screen_Y == 8'd50) ||
    (screen_X == 9'd270 && screen_Y == 8'd50) ||
    (screen_X == 9'd271 && screen_Y == 8'd50) ||
    (screen_X == 9'd272 && screen_Y == 8'd50) ||
    (screen_X == 9'd273 && screen_Y == 8'd50) ||
    (screen_X == 9'd274 && screen_Y == 8'd50) ||
    (screen_X == 9'd275 && screen_Y == 8'd50) ||
    (screen_X == 9'd276 && screen_Y == 8'd50) ||
    (screen_X == 9'd277 && screen_Y == 8'd50) ||
```



(screen\_X == 9'd278 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd279 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd280 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd281 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd282 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd283 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd284 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd285 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd286 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd287 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd288 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd289 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd290 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd291 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd292 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd293 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd294 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd295 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd296 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd297 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd298 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd299 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd300 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd301 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd302 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd303 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd304 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd305 && screen\_Y == 8'd50) ||

(screen\_X == 9'd306 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd307 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd308 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd252 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd253 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd254 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd255 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd256 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd258 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd259 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd260 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd261 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd262 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd263 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd264 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd265 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd266 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd267 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd268 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd269 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd270 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd271 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd272 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd273 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd274 && screen\_Y == 8'd108) ||

(screen\_X == 9'd275 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd276 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd277 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd278 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd279 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd280 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd281 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd282 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd283 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd284 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd285 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd286 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd287 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd288 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd289 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd290 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd291 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd292 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd293 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd294 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd295 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd296 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd297 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd298 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd299 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd300 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd301 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd302 && screen\_Y == 8'd108) ||

(screen\_X == 9'd303 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd304 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd305 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd306 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd307 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd308 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd51) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd52) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd53) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd54) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd56) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd57) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd58) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd59) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd60) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd61) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd62) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd63) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd64) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd65) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd66) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd67) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd68) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd69) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd70) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd71) ||

(screen\_X == 9'd251 && screen\_Y == 8'd72) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd73) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd74) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd75) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd76) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd77) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd78) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd79) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd80) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd81) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd82) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd83) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd84) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd85) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd86) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd87) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd88) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd90) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd91) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd92) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd93) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd94) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd95) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd96) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd98) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd99) ||

(screen\_X == 9'd251 && screen\_Y == 8'd100) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd101) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd102) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd103) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd104) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd105) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd106) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd107) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd51) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd52) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd53) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd54) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd56) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd57) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd58) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd59) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd60) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd61) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd62) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd63) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd64) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd65) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd66) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd67) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd68) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd69) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd70) ||

(screen\_X == 9'd309 && screen\_Y == 8'd71) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd72) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd73) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd74) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd75) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd76) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd77) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd78) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd79) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd80) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd81) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd82) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd83) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd84) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd85) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd86) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd87) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd88) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd90) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd91) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd92) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd93) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd94) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd95) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd96) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd98) ||

```

(screen_X == 9'd309 && screen_Y == 8'd99) ||
(screen_X == 9'd309 && screen_Y == 8'd100) ||
(screen_X == 9'd309 && screen_Y == 8'd101) ||
(screen_X == 9'd309 && screen_Y == 8'd102) ||
(screen_X == 9'd309 && screen_Y == 8'd103) ||
(screen_X == 9'd309 && screen_Y == 8'd104) ||
(screen_X == 9'd309 && screen_Y == 8'd105) ||
(screen_X == 9'd309 && screen_Y == 8'd106) ||
(screen_X == 9'd309 && screen_Y == 8'd107)
) begin // box border

    pixel_colour = colourful ? (((randNum_12b[6:4] ^
randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[6:4] ^ randNum_12b[4:2])) :
3'b111;

    end

    if (screen_X >= 9'd252 && screen_X <= 9'd308 && screen_Y
>= 8'd50 && screen_Y <= 8'd106) begin

        effective_X = screen_X - 9'd252;
        effective_Y = screen_Y - 8'd51;

        if (highscore == 12'd0) begin
            end

            if (highscore == 12'd2) begin

                if((effective_X == 9'd43 && effective_Y ==
8'd18) || (effective_X == 9'd43 && effective_Y == 8'd19) || (effective_X == 9'd43 &&
effective_Y == 8'd27) || (effective_X == 9'd43 && effective_Y == 8'd28) || (effective_X
== 9'd43 && effective_Y == 8'd29) || (effective_X == 9'd43 && effective_Y == 8'd30) ||
(effective_X == 9'd43 && effective_Y == 8'd31) || (effective_X == 9'd43 && effective_Y
== 8'd32) || (effective_X == 9'd43 && effective_Y == 8'd33) || (effective_X == 9'd43 &&
effective_Y == 8'd34) || (effective_X == 9'd43 && effective_Y == 8'd35) || (effective_X
== 9'd43 && effective_Y == 8'd36) || (effective_X == 9'd43 && effective_Y == 8'd37) ||
(effective_X == 9'd43 && effective_Y == 8'd38) ||

                    (effective_X == 9'd44 && effective_Y ==
8'd18) || (effective_X == 9'd44 && effective_Y == 8'd19) || (effective_X == 9'd44 &&
effective_Y == 8'd27) || (effective_X == 9'd44 && effective_Y == 8'd28) || (effective_X

```





```

                (effective_X == 9'd53 && effective_Y ==
8'd18) || (effective_X == 9'd53 && effective_Y == 8'd19) || (effective_X == 9'd53 &&
effective_Y == 8'd20) || (effective_X == 9'd53 && effective_Y == 8'd21) || (effective_X
== 9'd53 && effective_Y == 8'd22) || (effective_X == 9'd53 && effective_Y == 8'd23) ||
(effective_X == 9'd53 && effective_Y == 8'd24) || (effective_X == 9'd53 && effective_Y
== 8'd25) || (effective_X == 9'd53 && effective_Y == 8'd26) || (effective_X == 9'd53 &&
effective_Y == 8'd27) || (effective_X == 9'd53 && effective_Y == 8'd28) || (effective_X
== 9'd53 && effective_Y == 8'd37) || (effective_X == 9'd53 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd37) || (effective_X == 9'd54 && effective_Y == 8'd38)

```

```

            ) begin

```

```

                pixel_colour = colourful ?

```

```

                (((randNum_12b[7:5] ^ randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[7:5] ^
randNum_12b[4:2])) : 3'b111;

```

```

            end

```

```

        end

```

```

        if (highscore == 12'd4) begin

```

```

                if((effective_X == 9'd43 && effective_Y ==
8'd18) || (effective_X == 9'd43 && effective_Y == 8'd19) || (effective_X == 9'd43 &&
effective_Y == 8'd20) || (effective_X == 9'd43 && effective_Y == 8'd21) || (effective_X
== 9'd43 && effective_Y == 8'd22) || (effective_X == 9'd43 && effective_Y == 8'd23) ||
(effective_X == 9'd43 && effective_Y == 8'd24) || (effective_X == 9'd43 && effective_Y
== 8'd25) || (effective_X == 9'd43 && effective_Y == 8'd26) || (effective_X == 9'd43 &&
effective_Y == 8'd27) || (effective_X == 9'd43 && effective_Y == 8'd28) ||

```

```

                (effective_X == 9'd44 && effective_Y ==
8'd18) || (effective_X == 9'd44 && effective_Y == 8'd19) || (effective_X == 9'd44 &&
effective_Y == 8'd20) || (effective_X == 9'd44 && effective_Y == 8'd21) || (effective_X
== 9'd44 && effective_Y == 8'd22) || (effective_X == 9'd44 && effective_Y == 8'd23) ||
(effective_X == 9'd44 && effective_Y == 8'd24) || (effective_X == 9'd44 && effective_Y
== 8'd25) || (effective_X == 9'd44 && effective_Y == 8'd26) || (effective_X == 9'd44 &&
effective_Y == 8'd27) || (effective_X == 9'd44 && effective_Y == 8'd28) ||

```



```

== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

) begin

    pixel_colour = colourful ?
(((randNum_12b[10:8] ^ randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^
randNum_12b[4:2])) : 3'b111;

end

end

if (highscore == 12'd8) begin

    if((effective_X == 9'd43 && effective_Y ==
8'd18) || (effective_X == 9'd43 && effective_Y == 8'd19) || (effective_X == 9'd43 &&
effective_Y == 8'd20) || (effective_X == 9'd43 && effective_Y == 8'd21) || (effective_X
== 9'd43 && effective_Y == 8'd22) || (effective_X == 9'd43 && effective_Y == 8'd23) ||
(effective_X == 9'd43 && effective_Y == 8'd24) || (effective_X == 9'd43 && effective_Y
== 8'd25) || (effective_X == 9'd43 && effective_Y == 8'd26) || (effective_X == 9'd43 &&
effective_Y == 8'd27) || (effective_X == 9'd43 && effective_Y == 8'd28) || (effective_X
== 9'd43 && effective_Y == 8'd29) || (effective_X == 9'd43 && effective_Y == 8'd30) ||
(effective_X == 9'd43 && effective_Y == 8'd31) || (effective_X == 9'd43 && effective_Y
== 8'd32) || (effective_X == 9'd43 && effective_Y == 8'd33) || (effective_X == 9'd43 &&
effective_Y == 8'd34) || (effective_X == 9'd43 && effective_Y == 8'd35) || (effective_X
== 9'd43 && effective_Y == 8'd36) || (effective_X == 9'd43 && effective_Y == 8'd37) ||
(effective_X == 9'd43 && effective_Y == 8'd38) ||

(effective_X == 9'd44 && effective_Y ==
8'd18) || (effective_X == 9'd44 && effective_Y == 8'd19) || (effective_X == 9'd44 &&
effective_Y == 8'd20) || (effective_X == 9'd44 && effective_Y == 8'd21) || (effective_X
== 9'd44 && effective_Y == 8'd22) || (effective_X == 9'd44 && effective_Y == 8'd23) ||
(effective_X == 9'd44 && effective_Y == 8'd24) || (effective_X == 9'd44 && effective_Y
== 8'd25) || (effective_X == 9'd44 && effective_Y == 8'd26) || (effective_X == 9'd44 &&
effective_Y == 8'd27) || (effective_X == 9'd44 && effective_Y == 8'd28) || (effective_X
== 9'd44 && effective_Y == 8'd29) || (effective_X == 9'd44 && effective_Y == 8'd30) ||
(effective_X == 9'd44 && effective_Y == 8'd31) || (effective_X == 9'd44 && effective_Y
== 8'd32) || (effective_X == 9'd44 && effective_Y == 8'd33) || (effective_X == 9'd44 &&
effective_Y == 8'd34) || (effective_X == 9'd44 && effective_Y == 8'd35) || (effective_X
== 9'd44 && effective_Y == 8'd36) || (effective_X == 9'd44 && effective_Y == 8'd37) ||
(effective_X == 9'd44 && effective_Y == 8'd38) ||

(effective_X == 9'd45 && effective_Y ==
8'd18) || (effective_X == 9'd45 && effective_Y == 8'd19) || (effective_X == 9'd45 &&

```



```

== 8'd32) || (effective_X == 9'd53 && effective_Y == 8'd33) || (effective_X == 9'd53 &&
effective_Y == 8'd34) || (effective_X == 9'd53 && effective_Y == 8'd35) || (effective_X
== 9'd53 && effective_Y == 8'd36) || (effective_X == 9'd53 && effective_Y == 8'd37) ||
(effective_X == 9'd53 && effective_Y == 8'd38) ||

(effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

) begin

pixel_colour = colourful ?
(((randNum_12b[10:8] ^ randNum_12b[3:1]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^
randNum_12b[3:1])) : 3'b111;

end

end

if (highscore == 12'd16) begin

if((effective_X == 9'd39 && effective_Y ==
8'd18) || (effective_X == 9'd39 && effective_Y == 8'd19) || (effective_X == 9'd39 &&
effective_Y == 8'd20) || (effective_X == 9'd39 && effective_Y == 8'd21) || (effective_X
== 9'd39 && effective_Y == 8'd22) || (effective_X == 9'd39 && effective_Y == 8'd23) ||
(effective_X == 9'd39 && effective_Y == 8'd24) || (effective_X == 9'd39 && effective_Y
== 8'd25) || (effective_X == 9'd39 && effective_Y == 8'd26) || (effective_X == 9'd39 &&
effective_Y == 8'd27) || (effective_X == 9'd39 && effective_Y == 8'd28) || (effective_X
== 9'd39 && effective_Y == 8'd29) || (effective_X == 9'd39 && effective_Y == 8'd30) ||
(effective_X == 9'd39 && effective_Y == 8'd31) || (effective_X == 9'd39 && effective_Y
== 8'd32) || (effective_X == 9'd39 && effective_Y == 8'd33) || (effective_X == 9'd39 &&
effective_Y == 8'd34) || (effective_X == 9'd39 && effective_Y == 8'd35) || (effective_X
== 9'd39 && effective_Y == 8'd36) || (effective_X == 9'd39 && effective_Y == 8'd37) ||
(effective_X == 9'd39 && effective_Y == 8'd38) ||

(effective_X == 9'd40 && effective_Y ==
8'd18) || (effective_X == 9'd40 && effective_Y == 8'd19) || (effective_X == 9'd40 &&

```







```

effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

        ) begin

        pixel_colour = colourful ?
(((randNum_12b[6:4] ^ randNum_12b[7:5]) == 3'b0) ? 3'b111 : (randNum_12b[6:4] ^
randNum_12b[7:5])) : 3'b111;

    end

end

if (highscore == 12'd32) begin

    if((effective_X == 9'd29 && effective_Y ==
8'd18) || (effective_X == 9'd29 && effective_Y == 8'd19) || (effective_X == 9'd29 &&
effective_Y == 8'd27) || (effective_X == 9'd29 && effective_Y == 8'd28) || (effective_X
== 9'd29 && effective_Y == 8'd37) || (effective_X == 9'd29 && effective_Y == 8'd38) ||

        (effective_X == 9'd30 && effective_Y ==
8'd18) || (effective_X == 9'd30 && effective_Y == 8'd19) || (effective_X == 9'd30 &&
effective_Y == 8'd27) || (effective_X == 9'd30 && effective_Y == 8'd28) || (effective_X
== 9'd30 && effective_Y == 8'd37) || (effective_X == 9'd30 && effective_Y == 8'd38) ||

        (effective_X == 9'd31 && effective_Y ==
8'd18) || (effective_X == 9'd31 && effective_Y == 8'd19) || (effective_X == 9'd31 &&
effective_Y == 8'd27) || (effective_X == 9'd31 && effective_Y == 8'd28) || (effective_X
== 9'd31 && effective_Y == 8'd37) || (effective_X == 9'd31 && effective_Y == 8'd38) ||

        (effective_X == 9'd32 && effective_Y ==
8'd18) || (effective_X == 9'd32 && effective_Y == 8'd19) || (effective_X == 9'd32 &&
effective_Y == 8'd27) || (effective_X == 9'd32 && effective_Y == 8'd28) || (effective_X
== 9'd32 && effective_Y == 8'd37) || (effective_X == 9'd32 && effective_Y == 8'd38) ||

        (effective_X == 9'd33 && effective_Y ==
8'd18) || (effective_X == 9'd33 && effective_Y == 8'd19) || (effective_X == 9'd33 &&
effective_Y == 8'd27) || (effective_X == 9'd33 && effective_Y == 8'd28) || (effective_X
== 9'd33 && effective_Y == 8'd37) || (effective_X == 9'd33 && effective_Y == 8'd38) ||

        (effective_X == 9'd34 && effective_Y ==
8'd18) || (effective_X == 9'd34 && effective_Y == 8'd19) || (effective_X == 9'd34 &&

```





```

                (effective_X == 9'd49 && effective_Y ==
8'd18) || (effective_X == 9'd49 && effective_Y == 8'd19) || (effective_X == 9'd49 &&
effective_Y == 8'd27) || (effective_X == 9'd49 && effective_Y == 8'd28) || (effective_X
== 9'd49 && effective_Y == 8'd37) || (effective_X == 9'd49 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd50 && effective_Y ==
8'd18) || (effective_X == 9'd50 && effective_Y == 8'd19) || (effective_X == 9'd50 &&
effective_Y == 8'd27) || (effective_X == 9'd50 && effective_Y == 8'd28) || (effective_X
== 9'd50 && effective_Y == 8'd37) || (effective_X == 9'd50 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd51 && effective_Y ==
8'd18) || (effective_X == 9'd51 && effective_Y == 8'd19) || (effective_X == 9'd51 &&
effective_Y == 8'd27) || (effective_X == 9'd51 && effective_Y == 8'd28) || (effective_X
== 9'd51 && effective_Y == 8'd37) || (effective_X == 9'd51 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd52 && effective_Y ==
8'd18) || (effective_X == 9'd52 && effective_Y == 8'd19) || (effective_X == 9'd52 &&
effective_Y == 8'd27) || (effective_X == 9'd52 && effective_Y == 8'd28) || (effective_X
== 9'd52 && effective_Y == 8'd37) || (effective_X == 9'd52 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd53 && effective_Y ==
8'd18) || (effective_X == 9'd53 && effective_Y == 8'd19) || (effective_X == 9'd53 &&
effective_Y == 8'd20) || (effective_X == 9'd53 && effective_Y == 8'd21) || (effective_X
== 9'd53 && effective_Y == 8'd22) || (effective_X == 9'd53 && effective_Y == 8'd23) ||
(effective_X == 9'd53 && effective_Y == 8'd24) || (effective_X == 9'd53 && effective_Y
== 8'd25) || (effective_X == 9'd53 && effective_Y == 8'd26) || (effective_X == 9'd53 &&
effective_Y == 8'd27) || (effective_X == 9'd53 && effective_Y == 8'd28) || (effective_X
== 9'd53 && effective_Y == 8'd37) || (effective_X == 9'd53 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd37) || (effective_X == 9'd54 && effective_Y == 8'd38)

```

```

            ) begin

```

```

                pixel_colour = colourful ?

```

```

                (((randNum_12b[4:2] ^ randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[4:2] ^
randNum_12b[4:2])) : 3'b111;

```

```

            end

```

```

        end

```







```

(effective_X == 9'd53 && effective_Y == 8'd31) || (effective_X == 9'd53 && effective_Y
== 8'd32) || (effective_X == 9'd53 && effective_Y == 8'd33) || (effective_X == 9'd53 &&
effective_Y == 8'd34) || (effective_X == 9'd53 && effective_Y == 8'd35) || (effective_X
== 9'd53 && effective_Y == 8'd36) || (effective_X == 9'd53 && effective_Y == 8'd37) ||
(effective_X == 9'd53 && effective_Y == 8'd38) ||

(effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

) begin

pixel_colour = colourful ?
(((randNum_12b[7:5] ^ randNum_12b[8:6]) == 3'b0) ? 3'b111 : (randNum_12b[7:5] ^
randNum_12b[8:6])) : 3'b111;

end

end

if (highscore == 12'd128) begin

if((effective_X == 9'd25 && effective_Y ==
8'd18) || (effective_X == 9'd25 && effective_Y == 8'd19) || (effective_X == 9'd25 &&
effective_Y == 8'd20) || (effective_X == 9'd25 && effective_Y == 8'd21) || (effective_X
== 9'd25 && effective_Y == 8'd22) || (effective_X == 9'd25 && effective_Y == 8'd23) ||
(effective_X == 9'd25 && effective_Y == 8'd24) || (effective_X == 9'd25 && effective_Y
== 8'd25) || (effective_X == 9'd25 && effective_Y == 8'd26) || (effective_X == 9'd25 &&
effective_Y == 8'd27) || (effective_X == 9'd25 && effective_Y == 8'd28) || (effective_X
== 9'd25 && effective_Y == 8'd29) || (effective_X == 9'd25 && effective_Y == 8'd30) ||
(effective_X == 9'd25 && effective_Y == 8'd31) || (effective_X == 9'd25 && effective_Y
== 8'd32) || (effective_X == 9'd25 && effective_Y == 8'd33) || (effective_X == 9'd25 &&
effective_Y == 8'd34) || (effective_X == 9'd25 && effective_Y == 8'd35) || (effective_X
== 9'd25 && effective_Y == 8'd36) || (effective_X == 9'd25 && effective_Y == 8'd37) ||
(effective_X == 9'd25 && effective_Y == 8'd38) ||

```











```

== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

        ) begin

                pixel_colour = colourful ?
(((randNum_12b[11:9] ^ randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^
randNum_12b[4:2])) : 3'b111;

        end

end

if (highscore == 12'd256) begin

        if((effective_X == 9'd15 && effective_Y ==
8'd18) || (effective_X == 9'd15 && effective_Y == 8'd19) || (effective_X == 9'd15 &&
effective_Y == 8'd27) || (effective_X == 9'd15 && effective_Y == 8'd28) || (effective_X
== 9'd15 && effective_Y == 8'd29) || (effective_X == 9'd15 && effective_Y == 8'd30) ||
(effective_X == 9'd15 && effective_Y == 8'd31) || (effective_X == 9'd15 && effective_Y
== 8'd32) || (effective_X == 9'd15 && effective_Y == 8'd33) || (effective_X == 9'd15 &&
effective_Y == 8'd34) || (effective_X == 9'd15 && effective_Y == 8'd35) || (effective_X
== 9'd15 && effective_Y == 8'd36) || (effective_X == 9'd15 && effective_Y == 8'd37) ||
(effective_X == 9'd15 && effective_Y == 8'd38) ||

                (effective_X == 9'd16 && effective_Y ==
8'd18) || (effective_X == 9'd16 && effective_Y == 8'd19) || (effective_X == 9'd16 &&
effective_Y == 8'd27) || (effective_X == 9'd16 && effective_Y == 8'd28) || (effective_X
== 9'd16 && effective_Y == 8'd29) || (effective_X == 9'd16 && effective_Y == 8'd30) ||
(effective_X == 9'd16 && effective_Y == 8'd31) || (effective_X == 9'd16 && effective_Y
== 8'd32) || (effective_X == 9'd16 && effective_Y == 8'd33) || (effective_X == 9'd16 &&
effective_Y == 8'd34) || (effective_X == 9'd16 && effective_Y == 8'd35) || (effective_X
== 9'd16 && effective_Y == 8'd36) || (effective_X == 9'd16 && effective_Y == 8'd37) ||
(effective_X == 9'd16 && effective_Y == 8'd38) ||

                (effective_X == 9'd17 && effective_Y ==
8'd18) || (effective_X == 9'd17 && effective_Y == 8'd19) || (effective_X == 9'd17 &&
effective_Y == 8'd27) || (effective_X == 9'd17 && effective_Y == 8'd28) || (effective_X
== 9'd17 && effective_Y == 8'd37) || (effective_X == 9'd17 && effective_Y == 8'd38) ||

                (effective_X == 9'd18 && effective_Y ==
8'd18) || (effective_X == 9'd18 && effective_Y == 8'd19) || (effective_X == 9'd18 &&

```











```

effective_Y == 8'd27) || (effective_X == 9'd50 && effective_Y == 8'd28) || (effective_X
== 9'd50 && effective_Y == 8'd37) || (effective_X == 9'd50 && effective_Y == 8'd38) ||

(effective_X == 9'd51 && effective_Y ==
8'd18) || (effective_X == 9'd51 && effective_Y == 8'd19) || (effective_X == 9'd51 &&
effective_Y == 8'd27) || (effective_X == 9'd51 && effective_Y == 8'd28) || (effective_X
== 9'd51 && effective_Y == 8'd37) || (effective_X == 9'd51 && effective_Y == 8'd38) ||

(effective_X == 9'd52 && effective_Y ==
8'd18) || (effective_X == 9'd52 && effective_Y == 8'd19) || (effective_X == 9'd52 &&
effective_Y == 8'd27) || (effective_X == 9'd52 && effective_Y == 8'd28) || (effective_X
== 9'd52 && effective_Y == 8'd37) || (effective_X == 9'd52 && effective_Y == 8'd38) ||

(effective_X == 9'd53 && effective_Y ==
8'd18) || (effective_X == 9'd53 && effective_Y == 8'd19) || (effective_X == 9'd53 &&
effective_Y == 8'd27) || (effective_X == 9'd53 && effective_Y == 8'd28) || (effective_X
== 9'd53 && effective_Y == 8'd29) || (effective_X == 9'd53 && effective_Y == 8'd30) ||
(effective_X == 9'd53 && effective_Y == 8'd31) || (effective_X == 9'd53 && effective_Y
== 8'd32) || (effective_X == 9'd53 && effective_Y == 8'd33) || (effective_X == 9'd53 &&
effective_Y == 8'd34) || (effective_X == 9'd53 && effective_Y == 8'd35) || (effective_X
== 9'd53 && effective_Y == 8'd36) || (effective_X == 9'd53 && effective_Y == 8'd37) ||
(effective_X == 9'd53 && effective_Y == 8'd38) ||

(effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

) begin

pixel_colour = colourful ?
(((randNum_12b[6:4] ^ randNum_12b[9:7]) == 3'b0) ? 3'b111 : (randNum_12b[6:4] ^
randNum_12b[9:7])) : 3'b111;

end

end

if (highscore == 12'd512) begin

if((effective_X == 9'd15 && effective_Y ==
8'd18) || (effective_X == 9'd15 && effective_Y == 8'd19) || (effective_X == 9'd15 &&
effective_Y == 8'd20) || (effective_X == 9'd15 && effective_Y == 8'd21) || (effective_X

```





















(effective\_X == 9'd49 && effective\_Y ==  
8'd27) || (effective\_X == 9'd49 && effective\_Y == 8'd28) ||

(effective\_X == 9'd50 && effective\_Y ==  
8'd27) || (effective\_X == 9'd50 && effective\_Y == 8'd28) ||

(effective\_X == 9'd51 && effective\_Y ==  
8'd27) || (effective\_X == 9'd51 && effective\_Y == 8'd28) ||

(effective\_X == 9'd52 && effective\_Y ==  
8'd27) || (effective\_X == 9'd52 && effective\_Y == 8'd28) ||

(effective\_X == 9'd53 && effective\_Y ==  
8'd18) || (effective\_X == 9'd53 && effective\_Y == 8'd19) || (effective\_X == 9'd53 &&  
effective\_Y == 8'd20) || (effective\_X == 9'd53 && effective\_Y == 8'd21) || (effective\_X  
== 9'd53 && effective\_Y == 8'd22) || (effective\_X == 9'd53 && effective\_Y == 8'd23) ||  
(effective\_X == 9'd53 && effective\_Y == 8'd24) || (effective\_X == 9'd53 && effective\_Y  
== 8'd25) || (effective\_X == 9'd53 && effective\_Y == 8'd26) || (effective\_X == 9'd53 &&  
effective\_Y == 8'd27) || (effective\_X == 9'd53 && effective\_Y == 8'd28) || (effective\_X  
== 9'd53 && effective\_Y == 8'd29) || (effective\_X == 9'd53 && effective\_Y == 8'd30) ||  
(effective\_X == 9'd53 && effective\_Y == 8'd31) || (effective\_X == 9'd53 && effective\_Y  
== 8'd32) || (effective\_X == 9'd53 && effective\_Y == 8'd33) || (effective\_X == 9'd53 &&  
effective\_Y == 8'd34) || (effective\_X == 9'd53 && effective\_Y == 8'd35) || (effective\_X  
== 9'd53 && effective\_Y == 8'd36) || (effective\_X == 9'd53 && effective\_Y == 8'd37) ||  
(effective\_X == 9'd53 && effective\_Y == 8'd38) ||

(effective\_X == 9'd54 && effective\_Y ==  
8'd18) || (effective\_X == 9'd54 && effective\_Y == 8'd19) || (effective\_X == 9'd54 &&  
effective\_Y == 8'd20) || (effective\_X == 9'd54 && effective\_Y == 8'd21) || (effective\_X  
== 9'd54 && effective\_Y == 8'd22) || (effective\_X == 9'd54 && effective\_Y == 8'd23) ||  
(effective\_X == 9'd54 && effective\_Y == 8'd24) || (effective\_X == 9'd54 && effective\_Y  
== 8'd25) || (effective\_X == 9'd54 && effective\_Y == 8'd26) || (effective\_X == 9'd54 &&  
effective\_Y == 8'd27) || (effective\_X == 9'd54 && effective\_Y == 8'd28) || (effective\_X  
== 9'd54 && effective\_Y == 8'd29) || (effective\_X == 9'd54 && effective\_Y == 8'd30) ||  
(effective\_X == 9'd54 && effective\_Y == 8'd31) || (effective\_X == 9'd54 && effective\_Y  
== 8'd32) || (effective\_X == 9'd54 && effective\_Y == 8'd33) || (effective\_X == 9'd54 &&  
effective\_Y == 8'd34) || (effective\_X == 9'd54 && effective\_Y == 8'd35) || (effective\_X  
== 9'd54 && effective\_Y == 8'd36) || (effective\_X == 9'd54 && effective\_Y == 8'd37) ||  
(effective\_X == 9'd54 && effective\_Y == 8'd38)

) begin

pixel\_colour = colourful ?

((randNum\_12b[10:8] ^ randNum\_12b[9:7]) == 3'b0) ? 3'b111 : (randNum\_12b[10:8] ^  
randNum\_12b[9:7]) : 3'b111;

end

```
if (highscore == 12'd2048) begin
```

```
(effective_X == 9'd2 && effective_Y ==
8'd18) || (effective_X == 9'd2 && effective_Y == 8'd19) || (effective_X == 9'd2 &&
effective_Y == 8'd27) || (effective_X == 9'd2 && effective_Y == 8'd28) || (effective_X ==
9'd2 && effective_Y == 8'd29) || (effective_X == 9'd2 && effective_Y == 8'd30) ||
(effective_X == 9'd2 && effective_Y == 8'd31) || (effective_X == 9'd2 && effective_Y ==
8'd32) || (effective_X == 9'd2 && effective_Y == 8'd33) || (effective_X == 9'd2 &&
effective_Y == 8'd34) || (effective_X == 9'd2 && effective_Y == 8'd35) || (effective_X ==
9'd2 && effective_Y == 8'd36) || (effective_X == 9'd2 && effective_Y == 8'd37) ||
(effective_X == 9'd2 && effective_Y == 8'd38) ||
```

```

                                (effective_X == 9'd4 && effective_Y ==
8'd18) || (effective_X == 9'd4 && effective_Y == 8'd19) || (effective_X == 9'd4 &&
effective_Y == 8'd27) || (effective_X == 9'd4 && effective_Y == 8'd28) || (effective_X ==
9'd4 && effective_Y == 8'd37) || (effective_X == 9'd4 && effective_Y == 8'd38) ||

```

```
(effective_X == 9'd6 && effective_Y ==
8'd18) || (effective_X == 9'd6 && effective_Y == 8'd19) || (effective_X == 9'd6 &&
effective_Y == 8'd27) || (effective_X == 9'd6 && effective_Y == 8'd28) || (effective_X ==
9'd6 && effective_Y == 8'd37) || (effective_X == 9'd6 && effective_Y == 8'd38) ||
```

149















```

effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

        ) begin

            pixel_colour = colourful ?
(((randNum_12b[10:8] ^ randNum_12b[6:4]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^
randNum_12b[6:4])) : 3'b111;

        end

    end

//          if (highscore == 12'd4096) begin
//              if(effective_X == 6'd1 && effective_Y == 6'd1 ||
//                  effective_X == 6'd2 && effective_Y ==
6'd2 ||
//                  effective_X == 6'd3 && effective_Y ==
6'd3 ||
//                  effective_X == 6'd4 && effective_Y ==
6'd4 ||
//                  effective_X == 6'd5 && effective_Y ==
6'd5 ||
//                  effective_X == 6'd6 && effective_Y ==
6'd6 ||
//                  effective_X == 6'd7 && effective_Y ==
6'd7 ||
//                  effective_X == 6'd8 && effective_Y ==
6'd8 ||
//                  effective_X == 6'd9 && effective_Y ==
6'd9 ||

```

```
//          effective_X == 6'd10 && effective_Y ==
6'd10 ||

//          effective_X == 6'd11 && effective_Y ==
6'd11 ||

//          effective_X == 6'd12 && effective_Y ==
6'd12) begin

//          pixel_colour = 3'b111;

//          end

//          end

//          end

        end

    end

else begin

    if ((screen_X == 9'd38 && screen_Y == 8'd24) ||

        (screen_X == 9'd38 && screen_Y == 8'd25) ||

        (screen_X == 9'd38 && screen_Y == 8'd26) ||

        (screen_X == 9'd38 && screen_Y == 8'd27) ||

        (screen_X == 9'd38 && screen_Y == 8'd28) ||

        (screen_X == 9'd38 && screen_Y == 8'd29) ||

        (screen_X == 9'd38 && screen_Y == 8'd30) ||

        (screen_X == 9'd38 && screen_Y == 8'd31) ||

        (screen_X == 9'd38 && screen_Y == 8'd32) ||

        (screen_X == 9'd38 && screen_Y == 8'd33) ||

        (screen_X == 9'd38 && screen_Y == 8'd34) ||

        (screen_X == 9'd38 && screen_Y == 8'd35) ||

        (screen_X == 9'd38 && screen_Y == 8'd36) ||

        (screen_X == 9'd38 && screen_Y == 8'd37) ||

        (screen_X == 9'd38 && screen_Y == 8'd38) ||

        (screen_X == 9'd38 && screen_Y == 8'd39) ||

        (screen_X == 9'd38 && screen_Y == 8'd40) ||
```

(screen\_X == 9'd38 && screen\_Y == 8'd41) ||  
(screen\_X == 9'd38 && screen\_Y == 8'd42) ||  
(screen\_X == 9'd38 && screen\_Y == 8'd43) ||  
(screen\_X == 9'd38 && screen\_Y == 8'd44) ||  
(screen\_X == 9'd38 && screen\_Y == 8'd45) ||  
(screen\_X == 9'd38 && screen\_Y == 8'd46) ||  
(screen\_X == 9'd38 && screen\_Y == 8'd47) ||  
(screen\_X == 9'd38 && screen\_Y == 8'd48) ||  
(screen\_X == 9'd38 && screen\_Y == 8'd49) ||  
(screen\_X == 9'd38 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd38 && screen\_Y == 8'd51) ||  
(screen\_X == 9'd38 && screen\_Y == 8'd52) ||  
(screen\_X == 9'd38 && screen\_Y == 8'd53) ||  
(screen\_X == 9'd38 && screen\_Y == 8'd54) ||  
(screen\_X == 9'd38 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd39 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd40 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd41 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd42 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd43 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd44 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd45 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd46 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd47 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd48 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd49 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd50 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd51 && screen\_Y == 8'd55) ||

(screen\_X == 9'd52 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd53 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd54 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd55 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd56 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd57 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd58 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd59 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd60 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd61 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd62 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd63 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd64 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd65 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd39 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd40 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd41 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd42 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd43 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd44 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd45 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd46 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd47 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd48 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd49 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd50 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd51 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd52 && screen\_Y == 8'd97) ||

(screen\_X == 9'd53 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd54 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd55 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd56 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd57 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd58 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd59 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd60 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd61 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd62 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd63 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd64 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd65 && screen\_Y == 8'd97) ||

(screen\_X == 9'd66 && screen\_Y == 8'd24) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd25) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd26) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd27) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd28) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd29) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd30) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd31) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd32) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd33) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd34) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd35) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd36) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd37) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd38) ||



(screen\_X == 9'd66 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd40) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd41) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd42) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd43) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd44) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd45) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd46) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd47) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd48) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd49) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd51) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd52) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd53) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd54) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd56) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd57) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd58) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd59) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd60) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd61) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd62) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd63) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd64) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd65) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd66) ||

(screen\_X == 9'd66 && screen\_Y == 8'd67) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd68) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd69) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd70) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd71) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd72) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd73) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd74) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd75) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd76) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd77) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd78) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd79) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd80) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd81) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd82) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd83) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd84) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd85) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd86) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd87) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd88) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd90) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd91) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd92) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd93) ||  
(screen\_X == 9'd66 && screen\_Y == 8'd94) ||

```

(screen_X == 9'd66 && screen_Y == 8'd95) ||
(screen_X == 9'd66 && screen_Y == 8'd96) ||
(screen_X == 9'd66 && screen_Y == 8'd97)
) begin // Y
    pixel_colour = colourful ? (((randNum_12b[11:9] ^
randNum_12b[9:7]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^ randNum_12b[9:7])) :
3'b111;

```

```

end

```

```

if ((screen_X == 9'd90 && screen_Y == 8'd37) ||
    (screen_X == 9'd90 && screen_Y == 8'd38) ||
    (screen_X == 9'd90 && screen_Y == 8'd39) ||
    (screen_X == 9'd90 && screen_Y == 8'd40) ||
    (screen_X == 9'd90 && screen_Y == 8'd41) ||
    (screen_X == 9'd90 && screen_Y == 8'd42) ||
    (screen_X == 9'd90 && screen_Y == 8'd43) ||
    (screen_X == 9'd90 && screen_Y == 8'd44) ||
    (screen_X == 9'd90 && screen_Y == 8'd45) ||
    (screen_X == 9'd90 && screen_Y == 8'd46) ||
    (screen_X == 9'd90 && screen_Y == 8'd47) ||
    (screen_X == 9'd90 && screen_Y == 8'd48) ||
    (screen_X == 9'd90 && screen_Y == 8'd49) ||
    (screen_X == 9'd90 && screen_Y == 8'd50) ||
    (screen_X == 9'd90 && screen_Y == 8'd51) ||
    (screen_X == 9'd90 && screen_Y == 8'd52) ||
    (screen_X == 9'd90 && screen_Y == 8'd53) ||
    (screen_X == 9'd90 && screen_Y == 8'd54) ||
    (screen_X == 9'd90 && screen_Y == 8'd55) ||
    (screen_X == 9'd90 && screen_Y == 8'd56) ||
    (screen_X == 9'd90 && screen_Y == 8'd57) ||

```

(screen\_X == 9'd90 && screen\_Y == 8'd58) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd59) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd60) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd61) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd62) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd63) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd64) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd65) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd66) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd67) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd68) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd69) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd70) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd71) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd72) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd73) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd74) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd75) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd76) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd77) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd78) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd79) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd80) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd81) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd82) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd83) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd84) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd85) ||

(screen\_X == 9'd90 && screen\_Y == 8'd86) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd87) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd88) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd40) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd41) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd42) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd43) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd44) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd45) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd46) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd47) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd48) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd49) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd51) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd52) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd53) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd54) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd56) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd57) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd58) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd59) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd60) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd61) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd62) ||

(screen\_X == 9'd119 && screen\_Y == 8'd63) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd64) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd65) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd66) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd67) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd68) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd69) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd70) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd71) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd72) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd73) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd74) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd75) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd76) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd77) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd78) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd79) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd80) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd81) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd82) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd83) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd84) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd85) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd86) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd87) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd88) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd91 && screen\_Y == 8'd39) ||

(screen\_X == 9'd92 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd93 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd94 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd95 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd96 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd97 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd98 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd99 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd100 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd101 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd102 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd103 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd104 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd105 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd106 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd108 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd109 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd110 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd111 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd112 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd113 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd114 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd115 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd116 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd117 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd118 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd91 && screen\_Y == 8'd89) ||

```

(screen_X == 9'd92 && screen_Y == 8'd89) ||
(screen_X == 9'd93 && screen_Y == 8'd89) ||
(screen_X == 9'd94 && screen_Y == 8'd89) ||
(screen_X == 9'd95 && screen_Y == 8'd89) ||
(screen_X == 9'd96 && screen_Y == 8'd89) ||
(screen_X == 9'd97 && screen_Y == 8'd89) ||
(screen_X == 9'd98 && screen_Y == 8'd89) ||
(screen_X == 9'd99 && screen_Y == 8'd89) ||
(screen_X == 9'd100 && screen_Y == 8'd89) ||
(screen_X == 9'd101 && screen_Y == 8'd89) ||
(screen_X == 9'd102 && screen_Y == 8'd89) ||
(screen_X == 9'd103 && screen_Y == 8'd89) ||
(screen_X == 9'd104 && screen_Y == 8'd89) ||
(screen_X == 9'd105 && screen_Y == 8'd89) ||
(screen_X == 9'd106 && screen_Y == 8'd89) ||
(screen_X == 9'd107 && screen_Y == 8'd89) ||
(screen_X == 9'd108 && screen_Y == 8'd89) ||
(screen_X == 9'd109 && screen_Y == 8'd89) ||
(screen_X == 9'd110 && screen_Y == 8'd89) ||
(screen_X == 9'd111 && screen_Y == 8'd89) ||
(screen_X == 9'd112 && screen_Y == 8'd89) ||
(screen_X == 9'd113 && screen_Y == 8'd89) ||
(screen_X == 9'd114 && screen_Y == 8'd89) ||
(screen_X == 9'd115 && screen_Y == 8'd89) ||
(screen_X == 9'd116 && screen_Y == 8'd89) ||
(screen_X == 9'd117 && screen_Y == 8'd89) ||
(screen_X == 9'd118 && screen_Y == 8'd89)
) begin // O

```



```

        pixel_colour = colourful ? (((randNum_12b[10:8] ^
randNum_12b[6:4]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^ randNum_12b[6:4])) :
3'b111;

```

```

    end

```

```

    if ((screen_X == 9'd145 && screen_Y == 8'd37) ||
        (screen_X == 9'd145 && screen_Y == 8'd38) ||
        (screen_X == 9'd145 && screen_Y == 8'd39) ||
        (screen_X == 9'd145 && screen_Y == 8'd40) ||
        (screen_X == 9'd145 && screen_Y == 8'd41) ||
        (screen_X == 9'd145 && screen_Y == 8'd42) ||
        (screen_X == 9'd145 && screen_Y == 8'd43) ||
        (screen_X == 9'd145 && screen_Y == 8'd44) ||
        (screen_X == 9'd145 && screen_Y == 8'd45) ||
        (screen_X == 9'd145 && screen_Y == 8'd46) ||
        (screen_X == 9'd145 && screen_Y == 8'd47) ||
        (screen_X == 9'd145 && screen_Y == 8'd48) ||
        (screen_X == 9'd145 && screen_Y == 8'd49) ||
        (screen_X == 9'd145 && screen_Y == 8'd50) ||
        (screen_X == 9'd145 && screen_Y == 8'd51) ||
        (screen_X == 9'd145 && screen_Y == 8'd52) ||
        (screen_X == 9'd145 && screen_Y == 8'd53) ||
        (screen_X == 9'd145 && screen_Y == 8'd54) ||
        (screen_X == 9'd145 && screen_Y == 8'd55) ||
        (screen_X == 9'd145 && screen_Y == 8'd56) ||
        (screen_X == 9'd145 && screen_Y == 8'd57) ||
        (screen_X == 9'd145 && screen_Y == 8'd58) ||
        (screen_X == 9'd145 && screen_Y == 8'd59) ||
        (screen_X == 9'd145 && screen_Y == 8'd60) ||
        (screen_X == 9'd145 && screen_Y == 8'd61) ||

```

(screen\_X == 9'd145 && screen\_Y == 8'd62) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd63) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd64) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd65) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd66) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd67) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd68) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd69) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd70) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd71) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd72) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd73) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd74) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd75) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd76) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd77) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd78) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd79) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd80) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd81) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd82) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd83) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd84) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd85) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd86) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd87) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd88) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd89) ||

(screen\_X == 9'd182 && screen\_Y == 8'd37) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd38) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd39) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd40) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd41) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd42) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd43) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd44) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd45) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd46) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd47) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd48) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd49) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd51) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd52) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd53) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd54) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd56) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd57) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd58) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd59) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd60) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd61) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd62) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd63) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd64) ||

(screen\_X == 9'd182 && screen\_Y == 8'd65) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd66) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd67) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd68) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd69) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd70) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd71) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd72) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd73) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd74) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd75) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd76) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd77) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd78) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd79) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd80) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd81) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd82) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd83) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd84) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd85) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd86) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd87) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd88) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd90) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd91) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd92) ||

(screen\_X == 9'd182 && screen\_Y == 8'd93) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd94) ||  
(screen\_X == 9'd146 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd147 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd148 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd149 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd150 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd151 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd152 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd153 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd154 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd155 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd156 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd158 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd159 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd160 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd161 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd162 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd163 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd164 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd165 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd166 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd167 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd168 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd169 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd170 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd171 && screen\_Y == 8'd89) ||

```

(screen_X == 9'd172 && screen_Y == 8'd89) ||
(screen_X == 9'd173 && screen_Y == 8'd89) ||
(screen_X == 9'd174 && screen_Y == 8'd89) ||
(screen_X == 9'd175 && screen_Y == 8'd89) ||
(screen_X == 9'd176 && screen_Y == 8'd89) ||
(screen_X == 9'd177 && screen_Y == 8'd89) ||
(screen_X == 9'd178 && screen_Y == 8'd89) ||
(screen_X == 9'd179 && screen_Y == 8'd89) ||
(screen_X == 9'd180 && screen_Y == 8'd89) ||
(screen_X == 9'd181 && screen_Y == 8'd89)
) begin // U
    pixel_colour = colourful ? (((randNum_12b[10:8] ^
randNum_12b[2:0]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^ randNum_12b[2:0])) :
3'b111;
end

if ((screen_X == 9'd27 && screen_Y == 8'd120) ||
(screen_X == 9'd27 && screen_Y == 8'd121) ||
(screen_X == 9'd27 && screen_Y == 8'd122) ||
(screen_X == 9'd27 && screen_Y == 8'd123) ||
(screen_X == 9'd27 && screen_Y == 8'd124) ||
(screen_X == 9'd27 && screen_Y == 8'd125) ||
(screen_X == 9'd27 && screen_Y == 8'd126) ||
(screen_X == 9'd27 && screen_Y == 8'd127) ||
(screen_X == 9'd27 && screen_Y == 8'd128) ||
(screen_X == 9'd27 && screen_Y == 8'd129) ||
(screen_X == 9'd27 && screen_Y == 8'd130) ||
(screen_X == 9'd27 && screen_Y == 8'd131) ||
(screen_X == 9'd27 && screen_Y == 8'd132) ||
(screen_X == 9'd27 && screen_Y == 8'd133) ||

```

(screen\_X == 9'd27 && screen\_Y == 8'd134) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd135) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd136) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd138) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd139) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd140) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd141) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd142) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd143) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd145) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd146) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd147) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd148) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd149) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd150) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd151) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd152) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd153) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd154) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd155) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd156) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd157) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd158) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd159) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd160) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd161) ||

(screen\_X == 9'd27 && screen\_Y == 8'd162) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd163) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd164) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd165) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd166) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd167) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd168) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd169) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd171) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd172) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd173) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd174) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd175) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd176) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd177) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd178) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd180) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd181) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd182) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd183) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd184) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd185) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd186) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd187) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd188) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd189) ||



(screen\_X == 9'd27 && screen\_Y == 8'd190) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd191) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd192) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd193) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd194) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd195) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd196) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd198) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd199) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd200) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd202) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd203) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd204) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd205) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd206) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd207) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd208) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd209) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd210) ||  
(screen\_X == 9'd27 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd28 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd29 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd30 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd31 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd32 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd33 && screen\_Y == 8'd211) ||

(screen\_X == 9'd34 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd35 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd36 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd37 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd38 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd39 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd40 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd41 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd42 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd43 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd44 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd45 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd46 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd47 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd48 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd49 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd50 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd51 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd52 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd53 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd54 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd55 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd56 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd57 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd58 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd59 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd60 && screen\_Y == 8'd211) ||  
(screen\_X == 9'd61 && screen\_Y == 8'd211) ||

```

(screen_X == 9'd62 && screen_Y == 8'd211) ||
(screen_X == 9'd63 && screen_Y == 8'd211) ||
(screen_X == 9'd64 && screen_Y == 8'd211) ||
(screen_X == 9'd65 && screen_Y == 8'd211) ||
(screen_X == 9'd66 && screen_Y == 8'd211) ||
(screen_X == 9'd67 && screen_Y == 8'd211) ||
(screen_X == 9'd68 && screen_Y == 8'd211) ||
(screen_X == 9'd69 && screen_Y == 8'd211)
) begin // L
    pixel_colour = colourful ? (((randNum_12b[10:8] ^
randNum_12b[3:1]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^ randNum_12b[3:1])) :
3'b111;
end

if ((screen_X == 9'd71 && screen_Y == 8'd156) ||
(screen_X == 9'd72 && screen_Y == 8'd156) ||
(screen_X == 9'd73 && screen_Y == 8'd156) ||
(screen_X == 9'd74 && screen_Y == 8'd156) ||
(screen_X == 9'd75 && screen_Y == 8'd156) ||
(screen_X == 9'd76 && screen_Y == 8'd156) ||
(screen_X == 9'd77 && screen_Y == 8'd156) ||
(screen_X == 9'd78 && screen_Y == 8'd156) ||
(screen_X == 9'd79 && screen_Y == 8'd156) ||
(screen_X == 9'd80 && screen_Y == 8'd156) ||
(screen_X == 9'd81 && screen_Y == 8'd156) ||
(screen_X == 9'd82 && screen_Y == 8'd156) ||
(screen_X == 9'd83 && screen_Y == 8'd156) ||
(screen_X == 9'd84 && screen_Y == 8'd156) ||
(screen_X == 9'd85 && screen_Y == 8'd156) ||
(screen_X == 9'd86 && screen_Y == 8'd156) ||

```

```
(screen_X == 9'd87 && screen_Y == 8'd156) ||  
(screen_X == 9'd88 && screen_Y == 8'd156) ||  
(screen_X == 9'd89 && screen_Y == 8'd156) ||  
(screen_X == 9'd90 && screen_Y == 8'd156) ||  
(screen_X == 9'd91 && screen_Y == 8'd156) ||  
(screen_X == 9'd92 && screen_Y == 8'd156) ||  
(screen_X == 9'd93 && screen_Y == 8'd156) ||  
(screen_X == 9'd94 && screen_Y == 8'd156) ||  
(screen_X == 9'd95 && screen_Y == 8'd156) ||  
(screen_X == 9'd96 && screen_Y == 8'd156) ||  
(screen_X == 9'd97 && screen_Y == 8'd156) ||  
(screen_X == 9'd98 && screen_Y == 8'd156) ||  
(screen_X == 9'd99 && screen_Y == 8'd156) ||  
(screen_X == 9'd100 && screen_Y == 8'd156) ||  
(screen_X == 9'd101 && screen_Y == 8'd156) ||  
(screen_X == 9'd102 && screen_Y == 8'd156) ||  
(screen_X == 9'd103 && screen_Y == 8'd156) ||  
(screen_X == 9'd104 && screen_Y == 8'd156) ||  
(screen_X == 9'd105 && screen_Y == 8'd156) ||  
(screen_X == 9'd106 && screen_Y == 8'd156) ||  
(screen_X == 9'd107 && screen_Y == 8'd156) ||  
(screen_X == 9'd75 && screen_Y == 8'd197) ||  
(screen_X == 9'd76 && screen_Y == 8'd197) ||  
(screen_X == 9'd77 && screen_Y == 8'd197) ||  
(screen_X == 9'd78 && screen_Y == 8'd197) ||  
(screen_X == 9'd79 && screen_Y == 8'd197) ||  
(screen_X == 9'd80 && screen_Y == 8'd197) ||  
(screen_X == 9'd81 && screen_Y == 8'd197) ||
```

(screen\_X == 9'd82 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd83 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd84 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd85 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd86 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd87 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd88 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd89 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd90 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd91 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd92 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd93 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd94 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd95 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd96 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd97 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd98 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd99 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd100 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd101 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd102 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd103 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd104 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd105 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd106 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd157) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd158) ||

(screen\_X == 9'd75 && screen\_Y == 8'd159) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd160) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd161) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd162) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd163) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd164) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd165) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd166) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd167) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd168) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd169) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd171) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd172) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd173) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd174) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd175) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd176) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd177) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd178) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd180) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd181) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd182) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd183) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd184) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd185) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd186) ||

(screen\_X == 9'd75 && screen\_Y == 8'd187) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd188) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd189) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd190) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd191) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd192) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd193) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd194) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd195) ||  
(screen\_X == 9'd75 && screen\_Y == 8'd196) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd157) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd158) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd159) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd160) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd161) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd162) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd163) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd164) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd165) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd166) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd167) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd168) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd169) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd171) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd172) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd173) ||  
(screen\_X == 9'd107 && screen\_Y == 8'd174) ||

```

(screen_X == 9'd107 && screen_Y == 8'd175) ||
(screen_X == 9'd107 && screen_Y == 8'd176) ||
(screen_X == 9'd107 && screen_Y == 8'd177) ||
(screen_X == 9'd107 && screen_Y == 8'd178) ||
(screen_X == 9'd107 && screen_Y == 8'd179) ||
(screen_X == 9'd107 && screen_Y == 8'd180) ||
(screen_X == 9'd107 && screen_Y == 8'd181) ||
(screen_X == 9'd107 && screen_Y == 8'd182) ||
(screen_X == 9'd107 && screen_Y == 8'd183) ||
(screen_X == 9'd107 && screen_Y == 8'd184) ||
(screen_X == 9'd107 && screen_Y == 8'd185) ||
(screen_X == 9'd107 && screen_Y == 8'd186) ||
(screen_X == 9'd107 && screen_Y == 8'd187) ||
(screen_X == 9'd107 && screen_Y == 8'd188) ||
(screen_X == 9'd107 && screen_Y == 8'd189) ||
(screen_X == 9'd107 && screen_Y == 8'd190) ||
(screen_X == 9'd107 && screen_Y == 8'd191) ||
(screen_X == 9'd107 && screen_Y == 8'd192) ||
(screen_X == 9'd107 && screen_Y == 8'd193) ||
(screen_X == 9'd107 && screen_Y == 8'd194) ||
(screen_X == 9'd107 && screen_Y == 8'd195) ||
(screen_X == 9'd107 && screen_Y == 8'd196)
) begin // O
    pixel_colour = colourful ? (((randNum_12b[10:8] ^
randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^ randNum_12b[4:2])) :
3'b111;

end

if ((screen_X == 9'd125 && screen_Y == 8'd144) ||
    (screen_X == 9'd126 && screen_Y == 8'd144) ||

```



(screen\_X == 9'd127 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd128 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd129 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd130 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd131 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd132 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd133 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd134 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd135 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd136 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd137 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd138 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd139 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd140 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd141 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd142 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd143 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd144 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd146 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd147 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd148 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd149 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd150 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd151 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd152 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd153 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd154 && screen\_Y == 8'd144) ||

(screen\_X == 9'd155 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd156 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd158 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd126 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd127 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd128 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd129 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd130 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd131 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd132 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd133 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd134 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd135 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd136 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd137 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd138 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd139 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd140 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd141 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd142 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd143 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd144 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd146 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd147 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd148 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd149 && screen\_Y == 8'd179) ||

(screen\_X == 9'd150 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd151 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd152 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd153 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd154 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd155 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd156 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd108 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd109 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd110 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd111 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd112 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd113 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd114 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd115 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd116 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd117 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd118 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd120 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd121 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd122 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd123 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd124 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd126 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd127 && screen\_Y == 8'd212) ||

(screen\_X == 9'd128 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd129 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd130 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd131 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd132 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd133 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd134 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd135 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd136 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd137 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd138 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd139 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd140 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd141 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd142 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd143 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd144 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd145 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd146 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd147 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd148 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd149 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd150 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd151 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd152 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd153 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd154 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd155 && screen\_Y == 8'd212) ||

(screen\_X == 9'd156 && screen\_Y == 8'd212) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd145) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd146) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd147) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd148) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd149) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd150) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd151) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd152) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd153) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd154) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd155) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd156) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd157) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd158) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd159) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd160) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd161) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd162) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd163) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd164) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd165) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd166) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd167) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd168) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd169) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd171) ||

(screen\_X == 9'd125 && screen\_Y == 8'd172) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd173) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd174) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd175) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd176) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd177) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd178) ||  
(screen\_X == 9'd125 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd180) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd181) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd182) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd183) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd184) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd185) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd186) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd187) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd188) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd189) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd190) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd191) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd192) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd193) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd194) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd195) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd196) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd198) ||  
(screen\_X == 9'd157 && screen\_Y == 8'd199) ||

```

(screen_X == 9'd157 && screen_Y == 8'd200) ||
(screen_X == 9'd157 && screen_Y == 8'd201) ||
(screen_X == 9'd157 && screen_Y == 8'd202) ||
(screen_X == 9'd157 && screen_Y == 8'd203) ||
(screen_X == 9'd157 && screen_Y == 8'd204) ||
(screen_X == 9'd157 && screen_Y == 8'd205) ||
(screen_X == 9'd157 && screen_Y == 8'd206) ||
(screen_X == 9'd157 && screen_Y == 8'd207) ||
(screen_X == 9'd157 && screen_Y == 8'd208) ||
(screen_X == 9'd157 && screen_Y == 8'd209) ||
(screen_X == 9'd157 && screen_Y == 8'd210) ||
(screen_X == 9'd157 && screen_Y == 8'd211) ||
(screen_X == 9'd157 && screen_Y == 8'd212)
) begin // S

    pixel_colour = colourful ? (((randNum_12b[10:8] ^
randNum_12b[5:3]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^ randNum_12b[5:3])) :
3'b111;

end

if ((screen_X == 9'd175 && screen_Y == 8'd136) ||
(screen_X == 9'd175 && screen_Y == 8'd137) ||
(screen_X == 9'd175 && screen_Y == 8'd138) ||
(screen_X == 9'd175 && screen_Y == 8'd139) ||
(screen_X == 9'd175 && screen_Y == 8'd140) ||
(screen_X == 9'd175 && screen_Y == 8'd141) ||
(screen_X == 9'd175 && screen_Y == 8'd142) ||
(screen_X == 9'd175 && screen_Y == 8'd143) ||
(screen_X == 9'd175 && screen_Y == 8'd144) ||
(screen_X == 9'd175 && screen_Y == 8'd145) ||
(screen_X == 9'd175 && screen_Y == 8'd146) ||

```

(screen\_X == 9'd175 && screen\_Y == 8'd147) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd148) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd149) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd150) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd151) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd152) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd153) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd154) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd155) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd156) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd157) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd158) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd159) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd160) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd161) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd162) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd163) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd164) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd165) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd166) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd167) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd168) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd169) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd171) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd172) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd173) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd174) ||



(screen\_X == 9'd175 && screen\_Y == 8'd175) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd176) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd177) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd178) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd180) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd181) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd182) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd183) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd184) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd185) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd186) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd187) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd188) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd189) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd190) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd191) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd192) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd193) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd194) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd195) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd196) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd198) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd199) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd200) ||  
(screen\_X == 9'd175 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd176 && screen\_Y == 8'd170) ||

(screen\_X == 9'd177 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd178 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd179 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd180 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd181 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd183 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd184 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd185 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd186 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd187 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd188 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd189 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd190 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd191 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd192 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd193 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd194 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd195 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd196 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd176 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd177 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd178 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd179 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd180 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd181 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd183 && screen\_Y == 8'd137) ||

(screen\_X == 9'd184 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd185 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd186 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd187 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd188 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd189 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd190 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd191 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd192 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd193 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd194 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd195 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd196 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd197 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd198 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd199 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd200 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd201 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd202 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd203 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd204 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd205 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd206 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd207 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd208 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd209 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd210 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd211 && screen\_Y == 8'd137) ||

(screen\_X == 9'd212 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd213 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd214 && screen\_Y == 8'd137) ||  
(screen\_X == 9'd176 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd177 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd178 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd179 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd180 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd181 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd182 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd183 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd184 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd185 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd186 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd187 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd188 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd189 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd190 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd191 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd192 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd193 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd194 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd195 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd196 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd197 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd198 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd199 && screen\_Y == 8'd201) ||  
(screen\_X == 9'd200 && screen\_Y == 8'd201) ||

```

(screen_X == 9'd201 && screen_Y == 8'd201) ||
(screen_X == 9'd202 && screen_Y == 8'd201) ||
(screen_X == 9'd203 && screen_Y == 8'd201) ||
(screen_X == 9'd204 && screen_Y == 8'd201) ||
(screen_X == 9'd205 && screen_Y == 8'd201) ||
(screen_X == 9'd206 && screen_Y == 8'd201) ||
(screen_X == 9'd207 && screen_Y == 8'd201) ||
(screen_X == 9'd208 && screen_Y == 8'd201) ||
(screen_X == 9'd209 && screen_Y == 8'd201) ||
(screen_X == 9'd210 && screen_Y == 8'd201) ||
(screen_X == 9'd211 && screen_Y == 8'd201) ||
(screen_X == 9'd212 && screen_Y == 8'd201) ||
(screen_X == 9'd213 && screen_Y == 8'd201) ||
(screen_X == 9'd214 && screen_Y == 8'd201)
) begin // E

    pixel_colour = colourful ? (((randNum_12b[10:8] ^
randNum_12b[7:5]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^ randNum_12b[7:5])) :
3'b111;

end

if ((screen_X == 9'd238 && screen_Y == 8'd130) ||
    (screen_X == 9'd238 && screen_Y == 8'd131) ||
    (screen_X == 9'd238 && screen_Y == 8'd132) ||
    (screen_X == 9'd238 && screen_Y == 8'd133) ||
    (screen_X == 9'd238 && screen_Y == 8'd134) ||
    (screen_X == 9'd238 && screen_Y == 8'd135) ||
    (screen_X == 9'd238 && screen_Y == 8'd136) ||
    (screen_X == 9'd238 && screen_Y == 8'd137) ||
    (screen_X == 9'd238 && screen_Y == 8'd138) ||
    (screen_X == 9'd238 && screen_Y == 8'd139) ||

```

(screen\_X == 9'd238 && screen\_Y == 8'd140) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd141) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd142) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd143) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd145) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd146) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd147) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd148) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd149) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd150) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd151) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd152) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd153) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd154) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd155) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd156) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd157) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd158) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd159) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd160) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd161) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd162) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd163) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd164) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd165) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd166) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd167) ||

(screen\_X == 9'd238 && screen\_Y == 8'd168) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd169) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd171) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd172) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd173) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd174) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd175) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd176) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd177) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd178) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd180) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd181) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd182) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd183) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd184) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd185) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd186) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd187) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd188) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd189) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd190) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd191) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd192) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd193) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd194) ||  
(screen\_X == 9'd238 && screen\_Y == 8'd195) ||

```

(screen_X == 9'd238 && screen_Y == 8'd196) ||
(screen_X == 9'd238 && screen_Y == 8'd197) ||
(screen_X == 9'd238 && screen_Y == 8'd198) ||
(screen_X == 9'd238 && screen_Y == 8'd199) ||
(screen_X == 9'd238 && screen_Y == 8'd200) ||
(screen_X == 9'd238 && screen_Y == 8'd201) ||
(screen_X == 9'd238 && screen_Y == 8'd206) ||
(screen_X == 9'd238 && screen_Y == 8'd207)
) begin // !
    pixel_colour = colourful ? (((randNum_12b[9:7] ^
randNum_12b[2:0]) == 3'b0) ? 3'b111 : (randNum_12b[9:7] ^ randNum_12b[2:0])) :
3'b111;

```

```

end

```

```

if ((screen_X == 9'd257 && screen_Y == 8'd128) ||
    (screen_X == 9'd257 && screen_Y == 8'd129) ||
    (screen_X == 9'd257 && screen_Y == 8'd130) ||
    (screen_X == 9'd257 && screen_Y == 8'd131) ||
    (screen_X == 9'd257 && screen_Y == 8'd132) ||
    (screen_X == 9'd257 && screen_Y == 8'd133) ||
    (screen_X == 9'd257 && screen_Y == 8'd134) ||
    (screen_X == 9'd257 && screen_Y == 8'd135) ||
    (screen_X == 9'd257 && screen_Y == 8'd136) ||
    (screen_X == 9'd257 && screen_Y == 8'd137) ||
    (screen_X == 9'd257 && screen_Y == 8'd138) ||
    (screen_X == 9'd257 && screen_Y == 8'd139) ||
    (screen_X == 9'd257 && screen_Y == 8'd140) ||
    (screen_X == 9'd257 && screen_Y == 8'd141) ||
    (screen_X == 9'd257 && screen_Y == 8'd142) ||
    (screen_X == 9'd257 && screen_Y == 8'd143) ||

```



(screen\_X == 9'd257 && screen\_Y == 8'd144) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd145) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd146) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd147) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd148) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd149) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd150) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd151) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd152) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd153) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd154) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd155) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd156) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd157) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd158) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd159) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd160) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd161) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd162) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd163) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd164) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd165) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd166) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd167) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd168) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd169) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd170) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd171) ||

(screen\_X == 9'd257 && screen\_Y == 8'd172) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd173) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd174) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd175) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd176) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd177) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd178) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd179) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd180) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd181) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd182) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd183) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd184) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd185) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd186) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd187) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd188) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd189) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd190) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd191) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd192) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd193) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd194) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd195) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd196) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd197) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd198) ||  
(screen\_X == 9'd257 && screen\_Y == 8'd199) ||

```

(screen_X == 9'd257 && screen_Y == 8'd200) ||
(screen_X == 9'd257 && screen_Y == 8'd201) ||
(screen_X == 9'd257 && screen_Y == 8'd205) ||
(screen_X == 9'd257 && screen_Y == 8'd206) ||
(screen_X == 9'd257 && screen_Y == 8'd207)
) begin // !
    pixel_colour = colourful ? (((randNum_12b[9:7] ^
randNum_12b[3:1]) == 3'b0) ? 3'b111 : (randNum_12b[9:7] ^ randNum_12b[3:1])) :
3'b111;

end

if ((screen_X == 9'd27 && screen_Y == 8'd224) ||
    (screen_X == 9'd27 && screen_Y == 8'd225) ||
    (screen_X == 9'd27 && screen_Y == 8'd226) ||
    (screen_X == 9'd27 && screen_Y == 8'd227) ||
    (screen_X == 9'd27 && screen_Y == 8'd228) ||
    (screen_X == 9'd27 && screen_Y == 8'd229) ||
    (screen_X == 9'd27 && screen_Y == 8'd230) ||
    (screen_X == 9'd27 && screen_Y == 8'd231) ||
    (screen_X == 9'd27 && screen_Y == 8'd232) ||
    (screen_X == 9'd27 && screen_Y == 8'd233) ||
    (screen_X == 9'd27 && screen_Y == 8'd234) ||
    (screen_X == 9'd27 && screen_Y == 8'd235) ||
    (screen_X == 9'd32 && screen_Y == 8'd224) ||
    (screen_X == 9'd32 && screen_Y == 8'd225) ||
    (screen_X == 9'd32 && screen_Y == 8'd226) ||
    (screen_X == 9'd32 && screen_Y == 8'd227) ||
    (screen_X == 9'd32 && screen_Y == 8'd228) ||
    (screen_X == 9'd28 && screen_Y == 8'd224) ||
    (screen_X == 9'd29 && screen_Y == 8'd224) ||

```

```

        (screen_X == 9'd30 && screen_Y == 8'd224) ||
        (screen_X == 9'd31 && screen_Y == 8'd224) ||
        (screen_X == 9'd28 && screen_Y == 8'd228) ||
        (screen_X == 9'd29 && screen_Y == 8'd228) ||
        (screen_X == 9'd30 && screen_Y == 8'd228) ||
        (screen_X == 9'd31 && screen_Y == 8'd228)
    ) begin // P
        pixel_colour = colourful ? (((randNum_12b[9:7] ^
randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[9:7] ^ randNum_12b[4:2])) :
3'b111;

    end

    if ((screen_X == 9'd33 && screen_Y == 8'd230) ||
        (screen_X == 9'd33 && screen_Y == 8'd231) ||
        (screen_X == 9'd33 && screen_Y == 8'd232) ||
        (screen_X == 9'd33 && screen_Y == 8'd233) ||
        (screen_X == 9'd33 && screen_Y == 8'd234) ||
        (screen_X == 9'd33 && screen_Y == 8'd235) ||
        (screen_X == 9'd33 && screen_Y == 8'd236) ||
        (screen_X == 9'd34 && screen_Y == 8'd231) ||
        (screen_X == 9'd35 && screen_Y == 8'd231) ||
        (screen_X == 9'd36 && screen_Y == 8'd231) ||
        (screen_X == 9'd37 && screen_Y == 8'd231)
    ) begin // r
        pixel_colour = colourful ? (((randNum_12b[9:7] ^
randNum_12b[6:4]) == 3'b0) ? 3'b111 : (randNum_12b[9:7] ^ randNum_12b[6:4])) :
3'b111;

    end

    if ((screen_X == 9'd41 && screen_Y == 8'd227) ||
        (screen_X == 9'd41 && screen_Y == 8'd228) ||
        (screen_X == 9'd41 && screen_Y == 8'd229) ||

```

```

(screen_X == 9'd41 && screen_Y == 8'd230) ||
(screen_X == 9'd41 && screen_Y == 8'd231) ||
(screen_X == 9'd41 && screen_Y == 8'd232) ||
(screen_X == 9'd41 && screen_Y == 8'd233) ||
(screen_X == 9'd41 && screen_Y == 8'd234) ||
(screen_X == 9'd41 && screen_Y == 8'd235) ||
(screen_X == 9'd41 && screen_Y == 8'd236) ||
(screen_X == 9'd42 && screen_Y == 8'd227) ||
(screen_X == 9'd43 && screen_Y == 8'd227) ||
(screen_X == 9'd44 && screen_Y == 8'd227) ||
(screen_X == 9'd45 && screen_Y == 8'd227) ||
(screen_X == 9'd42 && screen_Y == 8'd236) ||
(screen_X == 9'd43 && screen_Y == 8'd236) ||
(screen_X == 9'd44 && screen_Y == 8'd236) ||
(screen_X == 9'd42 && screen_Y == 8'd231) ||
(screen_X == 9'd43 && screen_Y == 8'd231)
) begin // e

    pixel_colour = colourful ? (((randNum_12b[8:6] ^
randNum_12b[2:0]) == 3'b0) ? 3'b111 : (randNum_12b[8:6] ^ randNum_12b[2:0])) :
3'b111;

end

if ((screen_X == 9'd48 && screen_Y == 8'd227) ||
    (screen_X == 9'd49 && screen_Y == 8'd227) ||
    (screen_X == 9'd50 && screen_Y == 8'd227) ||
    (screen_X == 9'd48 && screen_Y == 8'd235) ||
    (screen_X == 9'd49 && screen_Y == 8'd235) ||
    (screen_X == 9'd50 && screen_Y == 8'd231) ||
    (screen_X == 9'd50 && screen_Y == 8'd232) ||
    (screen_X == 9'd50 && screen_Y == 8'd233) ||

```

```

(screen_X == 9'd50 && screen_Y == 8'd234) ||
(screen_X == 9'd50 && screen_Y == 8'd235) ||
(screen_X == 9'd48 && screen_Y == 8'd228) ||
(screen_X == 9'd48 && screen_Y == 8'd229) ||
(screen_X == 9'd48 && screen_Y == 8'd230) ||
(screen_X == 9'd48 && screen_Y == 8'd231) ||
(screen_X == 9'd49 && screen_Y == 8'd231)
) begin // s
    pixel_colour = colourful ? (((randNum_12b[8:6] ^
randNum_12b[3:1]) == 3'b0) ? 3'b111 : (randNum_12b[8:6] ^ randNum_12b[3:1])) :
3'b111;
end
if ((screen_X == 9'd53 && screen_Y == 8'd226) ||
    (screen_X == 9'd54 && screen_Y == 8'd226) ||
    (screen_X == 9'd55 && screen_Y == 8'd226) ||
    (screen_X == 9'd56 && screen_Y == 8'd226) ||
    (screen_X == 9'd52 && screen_Y == 8'd233) ||
    (screen_X == 9'd53 && screen_Y == 8'd233) ||
    (screen_X == 9'd54 && screen_Y == 8'd233) ||
    (screen_X == 9'd54 && screen_Y == 8'd229) ||
    (screen_X == 9'd53 && screen_Y == 8'd227) ||
    (screen_X == 9'd53 && screen_Y == 8'd228) ||
    (screen_X == 9'd53 && screen_Y == 8'd229) ||
    (screen_X == 9'd55 && screen_Y == 8'd229) ||
    (screen_X == 9'd55 && screen_Y == 8'd230) ||
    (screen_X == 9'd55 && screen_Y == 8'd231) ||
    (screen_X == 9'd55 && screen_Y == 8'd232) ||
    (screen_X == 9'd55 && screen_Y == 8'd233)
) begin // s

```

```

        pixel_colour = colourful ? (((randNum_12b[8:6] ^
randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[8:6] ^ randNum_12b[4:2])) :
3'b111;

```

```

    end

```

```

    if ((screen_X == 9'd63 && screen_Y == 8'd223) ||
        (screen_X == 9'd63 && screen_Y == 8'd224) ||
        (screen_X == 9'd63 && screen_Y == 8'd225) ||
        (screen_X == 9'd63 && screen_Y == 8'd226) ||
        (screen_X == 9'd63 && screen_Y == 8'd227) ||
        (screen_X == 9'd63 && screen_Y == 8'd228) ||
        (screen_X == 9'd63 && screen_Y == 8'd229) ||
        (screen_X == 9'd63 && screen_Y == 8'd230) ||
        (screen_X == 9'd63 && screen_Y == 8'd231) ||
        (screen_X == 9'd63 && screen_Y == 8'd232) ||
        (screen_X == 9'd63 && screen_Y == 8'd233) ||
        (screen_X == 9'd64 && screen_Y == 8'd227) ||
        (screen_X == 9'd65 && screen_Y == 8'd227) ||
        (screen_X == 9'd66 && screen_Y == 8'd227) ||
        (screen_X == 9'd64 && screen_Y == 8'd223) ||
        (screen_X == 9'd65 && screen_Y == 8'd223) ||
        (screen_X == 9'd66 && screen_Y == 8'd223) ||
        (screen_X == 9'd67 && screen_Y == 8'd223) ||
        (screen_X == 9'd64 && screen_Y == 8'd233) ||
        (screen_X == 9'd65 && screen_Y == 8'd233) ||
        (screen_X == 9'd66 && screen_Y == 8'd233) ||
        (screen_X == 9'd67 && screen_Y == 8'd233) ||
        (screen_X == 9'd68 && screen_Y == 8'd233)
    ) begin // E

```

```

        pixel_colour = colourful ? (((randNum_12b[8:6] ^
randNum_12b[7:5]) == 3'b0) ? 3'b111 : (randNum_12b[8:6] ^ randNum_12b[7:5])) :
3'b111;

```

```

    end

```

```

    if ((screen_X == 9'd71 && screen_Y == 8'd227) ||
        (screen_X == 9'd71 && screen_Y == 8'd228) ||
        (screen_X == 9'd71 && screen_Y == 8'd229) ||
        (screen_X == 9'd71 && screen_Y == 8'd230) ||
        (screen_X == 9'd71 && screen_Y == 8'd231) ||
        (screen_X == 9'd71 && screen_Y == 8'd232) ||
        (screen_X == 9'd71 && screen_Y == 8'd233) ||
        (screen_X == 9'd72 && screen_Y == 8'd228) ||
        (screen_X == 9'd73 && screen_Y == 8'd228) ||
        (screen_X == 9'd74 && screen_Y == 8'd228) ||
        (screen_X == 9'd75 && screen_Y == 8'd228) ||
        (screen_X == 9'd75 && screen_Y == 8'd229) ||
        (screen_X == 9'd75 && screen_Y == 8'd230) ||
        (screen_X == 9'd75 && screen_Y == 8'd231) ||
        (screen_X == 9'd75 && screen_Y == 8'd232) ||
        (screen_X == 9'd75 && screen_Y == 8'd233)
    ) begin // n

```

```

        pixel_colour = colourful ? (((randNum_12b[8:6] ^
randNum_12b[8:6]) == 3'b0) ? 3'b111 : (randNum_12b[8:6] ^ randNum_12b[8:6])) :
3'b111;

```

```

    end

```

```

    if ((screen_X == 9'd76 && screen_Y == 8'd224) ||
        (screen_X == 9'd77 && screen_Y == 8'd224) ||
        (screen_X == 9'd78 && screen_Y == 8'd224) ||
        (screen_X == 9'd79 && screen_Y == 8'd224) ||
        (screen_X == 9'd80 && screen_Y == 8'd224) ||

```



```

(screen_X == 9'd81 && screen_Y == 8'd224) ||
(screen_X == 9'd82 && screen_Y == 8'd224) ||
(screen_X == 9'd79 && screen_Y == 8'd225) ||
(screen_X == 9'd79 && screen_Y == 8'd226) ||
(screen_X == 9'd79 && screen_Y == 8'd227) ||
(screen_X == 9'd79 && screen_Y == 8'd228) ||
(screen_X == 9'd79 && screen_Y == 8'd229) ||
(screen_X == 9'd79 && screen_Y == 8'd230) ||
(screen_X == 9'd79 && screen_Y == 8'd231) ||
(screen_X == 9'd79 && screen_Y == 8'd232) ||
(screen_X == 9'd79 && screen_Y == 8'd233)
) begin // t
    pixel_colour = colourful ? (((randNum_12b[7:5] ^
randNum_12b[2:0]) == 3'b0) ? 3'b111 : (randNum_12b[7:5] ^ randNum_12b[2:0])) :
3'b111;
end
if ((screen_X == 9'd83 && screen_Y == 8'd225) ||
(screen_X == 9'd83 && screen_Y == 8'd226) ||
(screen_X == 9'd83 && screen_Y == 8'd227) ||
(screen_X == 9'd83 && screen_Y == 8'd228) ||
(screen_X == 9'd83 && screen_Y == 8'd229) ||
(screen_X == 9'd83 && screen_Y == 8'd230) ||
(screen_X == 9'd83 && screen_Y == 8'd231) ||
(screen_X == 9'd83 && screen_Y == 8'd232) ||
(screen_X == 9'd83 && screen_Y == 8'd233) ||
(screen_X == 9'd84 && screen_Y == 8'd225) ||
(screen_X == 9'd85 && screen_Y == 8'd225) ||
(screen_X == 9'd86 && screen_Y == 8'd225) ||
(screen_X == 9'd84 && screen_Y == 8'd230) ||

```

```

(screen_X == 9'd85 && screen_Y == 8'd230) ||
(screen_X == 9'd84 && screen_Y == 8'd233) ||
(screen_X == 9'd85 && screen_Y == 8'd233) ||
(screen_X == 9'd86 && screen_Y == 8'd233)
) begin // e
    pixel_colour = colourful ? (((randNum_12b[7:5] ^
randNum_12b[3:1]) == 3'b0) ? 3'b111 : (randNum_12b[7:5] ^ randNum_12b[3:1])) :
3'b111;

end

if ((screen_X == 9'd89 && screen_Y == 8'd223) ||
(screen_X == 9'd89 && screen_Y == 8'd224) ||
(screen_X == 9'd89 && screen_Y == 8'd225) ||
(screen_X == 9'd89 && screen_Y == 8'd226) ||
(screen_X == 9'd89 && screen_Y == 8'd227) ||
(screen_X == 9'd89 && screen_Y == 8'd228) ||
(screen_X == 9'd89 && screen_Y == 8'd229) ||
(screen_X == 9'd89 && screen_Y == 8'd230) ||
(screen_X == 9'd89 && screen_Y == 8'd231) ||
(screen_X == 9'd89 && screen_Y == 8'd232) ||
(screen_X == 9'd89 && screen_Y == 8'd233) ||
(screen_X == 9'd90 && screen_Y == 8'd223) ||
(screen_X == 9'd91 && screen_Y == 8'd223) ||
(screen_X == 9'd92 && screen_Y == 8'd223) ||
(screen_X == 9'd93 && screen_Y == 8'd223) ||
(screen_X == 9'd93 && screen_Y == 8'd224) ||
(screen_X == 9'd93 && screen_Y == 8'd225) ||
(screen_X == 9'd93 && screen_Y == 8'd226) ||
(screen_X == 9'd93 && screen_Y == 8'd227) ||
(screen_X == 9'd90 && screen_Y == 8'd227) ||

```

```

(screen_X == 9'd91 && screen_Y == 8'd227) ||
(screen_X == 9'd92 && screen_Y == 8'd227) ||
(screen_X == 9'd92 && screen_Y == 8'd228) ||
(screen_X == 9'd92 && screen_Y == 8'd229) ||
(screen_X == 9'd93 && screen_Y == 8'd230) ||
(screen_X == 9'd93 && screen_Y == 8'd231) ||
(screen_X == 9'd94 && screen_Y == 8'd232)
) begin // r
    pixel_colour = colourful ? (((randNum_12b[7:5] ^
randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[7:5] ^ randNum_12b[4:2])) :
3'b111;
end

if ((screen_X == 9'd100 && screen_Y == 8'd223) ||
    (screen_X == 9'd101 && screen_Y == 8'd223) ||
    (screen_X == 9'd102 && screen_Y == 8'd223) ||
    (screen_X == 9'd103 && screen_Y == 8'd223) ||
    (screen_X == 9'd104 && screen_Y == 8'd223) ||
    (screen_X == 9'd105 && screen_Y == 8'd223) ||
    (screen_X == 9'd106 && screen_Y == 8'd223) ||
    (screen_X == 9'd103 && screen_Y == 8'd224) ||
    (screen_X == 9'd103 && screen_Y == 8'd225) ||
    (screen_X == 9'd103 && screen_Y == 8'd226) ||
    (screen_X == 9'd103 && screen_Y == 8'd227) ||
    (screen_X == 9'd103 && screen_Y == 8'd228) ||
    (screen_X == 9'd103 && screen_Y == 8'd229) ||
    (screen_X == 9'd103 && screen_Y == 8'd230) ||
    (screen_X == 9'd103 && screen_Y == 8'd231) ||
    (screen_X == 9'd103 && screen_Y == 8'd232) ||
    (screen_X == 9'd103 && screen_Y == 8'd233)

```

```

        ) begin // T
            pixel_colour = colourful ? (((randNum_12b[7:5] ^
randNum_12b[5:3]) == 3'b0) ? 3'b111 : (randNum_12b[7:5] ^ randNum_12b[5:3])) :
3'b111;

        end

        if ((screen_X == 9'd107 && screen_Y == 8'd227) ||
            (screen_X == 9'd107 && screen_Y == 8'd228) ||
            (screen_X == 9'd107 && screen_Y == 8'd229) ||
            (screen_X == 9'd107 && screen_Y == 8'd230) ||
            (screen_X == 9'd107 && screen_Y == 8'd231) ||
            (screen_X == 9'd107 && screen_Y == 8'd232) ||
            (screen_X == 9'd111 && screen_Y == 8'd227) ||
            (screen_X == 9'd111 && screen_Y == 8'd228) ||
            (screen_X == 9'd111 && screen_Y == 8'd229) ||
            (screen_X == 9'd111 && screen_Y == 8'd230) ||
            (screen_X == 9'd111 && screen_Y == 8'd231) ||
            (screen_X == 9'd111 && screen_Y == 8'd232) ||
            (screen_X == 9'd107 && screen_Y == 8'd227) ||
            (screen_X == 9'd108 && screen_Y == 8'd227) ||
            (screen_X == 9'd109 && screen_Y == 8'd227) ||
            (screen_X == 9'd107 && screen_Y == 8'd232) ||
            (screen_X == 9'd108 && screen_Y == 8'd232) ||
            (screen_X == 9'd109 && screen_Y == 8'd232)
        ) begin // o
            pixel_colour = colourful ? (((randNum_12b[7:5] ^
randNum_12b[6:4]) == 3'b0) ? 3'b111 : (randNum_12b[7:5] ^ randNum_12b[6:4])) :
3'b111;

        end

        if ((screen_X == 9'd118 && screen_Y == 8'd222) ||
            (screen_X == 9'd118 && screen_Y == 8'd223) ||

```

(screen\_X == 9'd118 && screen\_Y == 8'd224) ||  
(screen\_X == 9'd118 && screen\_Y == 8'd225) ||  
(screen\_X == 9'd118 && screen\_Y == 8'd226) ||  
(screen\_X == 9'd118 && screen\_Y == 8'd227) ||  
(screen\_X == 9'd118 && screen\_Y == 8'd228) ||  
(screen\_X == 9'd118 && screen\_Y == 8'd229) ||  
(screen\_X == 9'd118 && screen\_Y == 8'd230) ||  
(screen\_X == 9'd118 && screen\_Y == 8'd231) ||  
(screen\_X == 9'd118 && screen\_Y == 8'd232) ||  
(screen\_X == 9'd118 && screen\_Y == 8'd233) ||  
(screen\_X == 9'd123 && screen\_Y == 8'd222) ||  
(screen\_X == 9'd123 && screen\_Y == 8'd223) ||  
(screen\_X == 9'd123 && screen\_Y == 8'd224) ||  
(screen\_X == 9'd123 && screen\_Y == 8'd225) ||  
(screen\_X == 9'd123 && screen\_Y == 8'd226) ||  
(screen\_X == 9'd121 && screen\_Y == 8'd228) ||  
(screen\_X == 9'd121 && screen\_Y == 8'd229) ||  
(screen\_X == 9'd121 && screen\_Y == 8'd230) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd222) ||  
(screen\_X == 9'd120 && screen\_Y == 8'd222) ||  
(screen\_X == 9'd121 && screen\_Y == 8'd222) ||  
(screen\_X == 9'd122 && screen\_Y == 8'd222) ||  
(screen\_X == 9'd119 && screen\_Y == 8'd226) ||  
(screen\_X == 9'd120 && screen\_Y == 8'd226) ||  
(screen\_X == 9'd121 && screen\_Y == 8'd226) ||  
(screen\_X == 9'd122 && screen\_Y == 8'd226) ||  
(screen\_X == 9'd122 && screen\_Y == 8'd231) ||  
(screen\_X == 9'd122 && screen\_Y == 8'd232) ||

```

(screen_X == 9'd120 && screen_Y == 8'd227)
) begin // R
    pixel_colour = colourful ? (((randNum_12b[7:5] ^
randNum_12b[7:5]) == 3'b0) ? 3'b111 : (randNum_12b[7:5] ^ randNum_12b[7:5])) :
3'b111;

end

if ((screen_X == 9'd125 && screen_Y == 8'd222) ||
    (screen_X == 9'd125 && screen_Y == 8'd223) ||
    (screen_X == 9'd125 && screen_Y == 8'd224) ||
    (screen_X == 9'd125 && screen_Y == 8'd225) ||
    (screen_X == 9'd125 && screen_Y == 8'd226) ||
    (screen_X == 9'd125 && screen_Y == 8'd227) ||
    (screen_X == 9'd125 && screen_Y == 8'd228) ||
    (screen_X == 9'd125 && screen_Y == 8'd229) ||
    (screen_X == 9'd125 && screen_Y == 8'd230) ||
    (screen_X == 9'd125 && screen_Y == 8'd231) ||
    (screen_X == 9'd126 && screen_Y == 8'd222) ||
    (screen_X == 9'd127 && screen_Y == 8'd222) ||
    (screen_X == 9'd128 && screen_Y == 8'd222) ||
    (screen_X == 9'd129 && screen_Y == 8'd222) ||
    (screen_X == 9'd126 && screen_Y == 8'd226) ||
    (screen_X == 9'd127 && screen_Y == 8'd226) ||
    (screen_X == 9'd126 && screen_Y == 8'd231) ||
    (screen_X == 9'd127 && screen_Y == 8'd231) ||
    (screen_X == 9'd128 && screen_Y == 8'd231) ||
    (screen_X == 9'd129 && screen_Y == 8'd231)
) begin // e
    pixel_colour = colourful ? (((randNum_12b[11:9] ^
randNum_12b[2:0]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^ randNum_12b[2:0])) :
3'b111;

```

```

end
if ((screen_X == 9'd131 && screen_Y == 8'd222) ||
    (screen_X == 9'd131 && screen_Y == 8'd223) ||
    (screen_X == 9'd131 && screen_Y == 8'd224) ||
    (screen_X == 9'd131 && screen_Y == 8'd225) ||
    (screen_X == 9'd131 && screen_Y == 8'd226) ||
    (screen_X == 9'd134 && screen_Y == 8'd226) ||
    (screen_X == 9'd134 && screen_Y == 8'd227) ||
    (screen_X == 9'd134 && screen_Y == 8'd228) ||
    (screen_X == 9'd134 && screen_Y == 8'd229) ||
    (screen_X == 9'd134 && screen_Y == 8'd230) ||
    (screen_X == 9'd134 && screen_Y == 8'd231) ||
    (screen_X == 9'd132 && screen_Y == 8'd222) ||
    (screen_X == 9'd133 && screen_Y == 8'd222) ||
    (screen_X == 9'd134 && screen_Y == 8'd222) ||
    (screen_X == 9'd131 && screen_Y == 8'd231) ||
    (screen_X == 9'd132 && screen_Y == 8'd231) ||
    (screen_X == 9'd133 && screen_Y == 8'd231) ||
    (screen_X == 9'd132 && screen_Y == 8'd226) ||
    (screen_X == 9'd133 && screen_Y == 8'd226)
) begin // s
    pixel_colour = colourful ? (((randNum_12b[11:9] ^
randNum_12b[11:9]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^ randNum_12b[11:9])) :
3'b111;
end
if ((screen_X == 9'd137 && screen_Y == 8'd222) ||
    (screen_X == 9'd138 && screen_Y == 8'd222) ||
    (screen_X == 9'd139 && screen_Y == 8'd222) ||
    (screen_X == 9'd136 && screen_Y == 8'd221) ||

```

```

(screen_X == 9'd136 && screen_Y == 8'd222) ||
(screen_X == 9'd136 && screen_Y == 8'd223) ||
(screen_X == 9'd136 && screen_Y == 8'd224) ||
(screen_X == 9'd136 && screen_Y == 8'd225) ||
(screen_X == 9'd136 && screen_Y == 8'd226) ||
(screen_X == 9'd136 && screen_Y == 8'd227) ||
(screen_X == 9'd136 && screen_Y == 8'd228) ||
(screen_X == 9'd136 && screen_Y == 8'd229) ||
(screen_X == 9'd136 && screen_Y == 8'd230) ||
(screen_X == 9'd136 && screen_Y == 8'd231) ||
(screen_X == 9'd137 && screen_Y == 8'd226) ||
(screen_X == 9'd138 && screen_Y == 8'd226) ||
(screen_X == 9'd139 && screen_Y == 8'd226) ||
(screen_X == 9'd137 && screen_Y == 8'd231) ||
(screen_X == 9'd138 && screen_Y == 8'd231) ||
(screen_X == 9'd139 && screen_Y == 8'd231) ||
(screen_X == 9'd140 && screen_Y == 8'd231)
) begin // e

    pixel_colour = colourful ? (((randNum_12b[11:9] ^
randNum_12b[3:1]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^ randNum_12b[3:1])) :
3'b111;

end

if ((screen_X == 9'd141 && screen_Y == 8'd222) ||
    (screen_X == 9'd142 && screen_Y == 8'd222) ||
    (screen_X == 9'd143 && screen_Y == 8'd222) ||
    (screen_X == 9'd144 && screen_Y == 8'd222) ||
    (screen_X == 9'd145 && screen_Y == 8'd222) ||
    (screen_X == 9'd143 && screen_Y == 8'd223) ||
    (screen_X == 9'd143 && screen_Y == 8'd224) ||

```



```

(screen_X == 9'd143 && screen_Y == 8'd225) ||
(screen_X == 9'd143 && screen_Y == 8'd226) ||
(screen_X == 9'd143 && screen_Y == 8'd227) ||
(screen_X == 9'd143 && screen_Y == 8'd228) ||
(screen_X == 9'd143 && screen_Y == 8'd229) ||
(screen_X == 9'd143 && screen_Y == 8'd230) ||
(screen_X == 9'd143 && screen_Y == 8'd231) ||
(screen_X == 9'd143 && screen_Y == 8'd232) ||
(screen_X == 9'd143 && screen_Y == 8'd233) ||
(screen_X == 9'd143 && screen_Y == 8'd234)
) begin // t
    pixel_colour = colourful ? (((randNum_12b[11:9] ^
randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^ randNum_12b[4:2])) :
3'b111;
end
if ((screen_X == 9'd207 && screen_Y == 8'd227) ||
(screen_X == 9'd208 && screen_Y == 8'd227) ||
(screen_X == 9'd209 && screen_Y == 8'd227) ||
(screen_X == 9'd210 && screen_Y == 8'd227) ||
(screen_X == 9'd211 && screen_Y == 8'd227) ||
(screen_X == 9'd212 && screen_Y == 8'd227) ||
(screen_X == 9'd209 && screen_Y == 8'd228) ||
(screen_X == 9'd209 && screen_Y == 8'd229) ||
(screen_X == 9'd209 && screen_Y == 8'd230) ||
(screen_X == 9'd209 && screen_Y == 8'd231) ||
(screen_X == 9'd209 && screen_Y == 8'd232) ||
(screen_X == 9'd209 && screen_Y == 8'd233) ||
(screen_X == 9'd209 && screen_Y == 8'd234) ||
(screen_X == 9'd209 && screen_Y == 8'd235)

```

```

) begin // T
    pixel_colour = colourful ? (((randNum_12b[11:9] ^
randNum_12b[5:3]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^ randNum_12b[5:3])) :
3'b111;

end

if ((screen_X == 9'd215 && screen_Y == 8'd227) ||
    (screen_X == 9'd215 && screen_Y == 8'd228) ||
    (screen_X == 9'd215 && screen_Y == 8'd229) ||
    (screen_X == 9'd215 && screen_Y == 8'd230) ||
    (screen_X == 9'd215 && screen_Y == 8'd231) ||
    (screen_X == 9'd215 && screen_Y == 8'd232) ||
    (screen_X == 9'd215 && screen_Y == 8'd233) ||
    (screen_X == 9'd215 && screen_Y == 8'd234) ||
    (screen_X == 9'd215 && screen_Y == 8'd235) ||
    (screen_X == 9'd215 && screen_Y == 8'd236) ||
    (screen_X == 9'd219 && screen_Y == 8'd226) ||
    (screen_X == 9'd219 && screen_Y == 8'd227) ||
    (screen_X == 9'd219 && screen_Y == 8'd228) ||
    (screen_X == 9'd219 && screen_Y == 8'd229) ||
    (screen_X == 9'd219 && screen_Y == 8'd230) ||
    (screen_X == 9'd219 && screen_Y == 8'd231) ||
    (screen_X == 9'd219 && screen_Y == 8'd232) ||
    (screen_X == 9'd219 && screen_Y == 8'd233) ||
    (screen_X == 9'd219 && screen_Y == 8'd234) ||
    (screen_X == 9'd219 && screen_Y == 8'd235) ||
    (screen_X == 9'd216 && screen_Y == 8'd232) ||
    (screen_X == 9'd217 && screen_Y == 8'd232) ||
    (screen_X == 9'd218 && screen_Y == 8'd232)
) begin // h

```

```

        pixel_colour = colourful ? (((randNum_12b[11:9] ^
randNum_12b[6:4]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^ randNum_12b[6:4])) :
3'b111;

```

```

    end

```

```

    if ((screen_X == 9'd222 && screen_Y == 8'd227) ||
        (screen_X == 9'd231 && screen_Y == 8'd227) ||
        (screen_X == 9'd223 && screen_Y == 8'd228) ||
        (screen_X == 9'd230 && screen_Y == 8'd228) ||
        (screen_X == 9'd224 && screen_Y == 8'd229) ||
        (screen_X == 9'd229 && screen_Y == 8'd229) ||
        (screen_X == 9'd225 && screen_Y == 8'd230) ||
        (screen_X == 9'd228 && screen_Y == 8'd230) ||
        (screen_X == 9'd226 && screen_Y == 8'd231) ||
        (screen_X == 9'd227 && screen_Y == 8'd231) ||
        (screen_X == 9'd225 && screen_Y == 8'd232) ||
        (screen_X == 9'd228 && screen_Y == 8'd232) ||
        (screen_X == 9'd224 && screen_Y == 8'd233) ||
        (screen_X == 9'd229 && screen_Y == 8'd233) ||
        (screen_X == 9'd223 && screen_Y == 8'd234) ||
        (screen_X == 9'd230 && screen_Y == 8'd234) ||
        (screen_X == 9'd222 && screen_Y == 8'd235) ||
        (screen_X == 9'd231 && screen_Y == 8'd235) ||
        (screen_X == 9'd221 && screen_Y == 8'd236)
    ) begin // x

```

```

        pixel_colour = colourful ? (((randNum_12b[11:9] ^
randNum_12b[7:5]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^ randNum_12b[7:5])) :
3'b111;

```

```

    end

```

```

    if ((screen_X == 9'd240 && screen_Y == 8'd226) ||
        (screen_X == 9'd240 && screen_Y == 8'd227) ||

```

```

(screen_X == 9'd240 && screen_Y == 8'd228) ||
(screen_X == 9'd240 && screen_Y == 8'd229) ||
(screen_X == 9'd240 && screen_Y == 8'd230) ||
(screen_X == 9'd240 && screen_Y == 8'd231) ||
(screen_X == 9'd240 && screen_Y == 8'd232) ||
(screen_X == 9'd240 && screen_Y == 8'd233) ||
(screen_X == 9'd240 && screen_Y == 8'd234) ||
(screen_X == 9'd240 && screen_Y == 8'd235) ||
(screen_X == 9'd240 && screen_Y == 8'd236) ||
(screen_X == 9'd241 && screen_Y == 8'd227) ||
(screen_X == 9'd242 && screen_Y == 8'd227) ||
(screen_X == 9'd243 && screen_Y == 8'd227) ||
(screen_X == 9'd244 && screen_Y == 8'd227) ||
(screen_X == 9'd245 && screen_Y == 8'd227) ||
(screen_X == 9'd246 && screen_Y == 8'd227) ||
(screen_X == 9'd241 && screen_Y == 8'd231) ||
(screen_X == 9'd242 && screen_Y == 8'd231) ||
(screen_X == 9'd243 && screen_Y == 8'd231)
) begin // F
    pixel_colour = colourful ? (((randNum_12b[11:9] ^
randNum_12b[8:6]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^ randNum_12b[8:6])) :
3'b111;
end
if ((screen_X == 9'd246 && screen_Y == 8'd230) ||
(screen_X == 9'd246 && screen_Y == 8'd231) ||
(screen_X == 9'd246 && screen_Y == 8'd232) ||
(screen_X == 9'd246 && screen_Y == 8'd233) ||
(screen_X == 9'd246 && screen_Y == 8'd234) ||
(screen_X == 9'd246 && screen_Y == 8'd235) ||

```

```

(screen_X == 9'd246 && screen_Y == 8'd236) ||
(screen_X == 9'd250 && screen_Y == 8'd230) ||
(screen_X == 9'd250 && screen_Y == 8'd231) ||
(screen_X == 9'd250 && screen_Y == 8'd232) ||
(screen_X == 9'd250 && screen_Y == 8'd233) ||
(screen_X == 9'd250 && screen_Y == 8'd234) ||
(screen_X == 9'd250 && screen_Y == 8'd235) ||
(screen_X == 9'd250 && screen_Y == 8'd236) ||
(screen_X == 9'd247 && screen_Y == 8'd230) ||
(screen_X == 9'd248 && screen_Y == 8'd230) ||
(screen_X == 9'd249 && screen_Y == 8'd230) ||
(screen_X == 9'd247 && screen_Y == 8'd236) ||
(screen_X == 9'd248 && screen_Y == 8'd236) ||
(screen_X == 9'd249 && screen_Y == 8'd236)
) begin // o
    pixel_colour = colourful ? (((randNum_12b[11:9] ^
randNum_12b[9:7]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^ randNum_12b[9:7])) :
3'b111;

end

if ((screen_X == 9'd253 && screen_Y == 8'd230) ||
    (screen_X == 9'd253 && screen_Y == 8'd231) ||
    (screen_X == 9'd253 && screen_Y == 8'd232) ||
    (screen_X == 9'd253 && screen_Y == 8'd233) ||
    (screen_X == 9'd253 && screen_Y == 8'd234) ||
    (screen_X == 9'd253 && screen_Y == 8'd235) ||
    (screen_X == 9'd254 && screen_Y == 8'd231) ||
    (screen_X == 9'd255 && screen_Y == 8'd231) ||
    (screen_X == 9'd256 && screen_Y == 8'd231) ||
    (screen_X == 9'd257 && screen_Y == 8'd231)

```

```

) begin // r
    pixel_colour = colourful ? (((randNum_12b[11:9] ^
randNum_12b[10:8]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^ randNum_12b[10:8])) :
3'b111;

end

if ((screen_X == 9'd266 && screen_Y == 8'd223) ||
    (screen_X == 9'd266 && screen_Y == 8'd224) ||
    (screen_X == 9'd266 && screen_Y == 8'd225) ||
    (screen_X == 9'd266 && screen_Y == 8'd226) ||
    (screen_X == 9'd266 && screen_Y == 8'd227) ||
    (screen_X == 9'd266 && screen_Y == 8'd228) ||
    (screen_X == 9'd266 && screen_Y == 8'd229) ||
    (screen_X == 9'd266 && screen_Y == 8'd230) ||
    (screen_X == 9'd266 && screen_Y == 8'd231) ||
    (screen_X == 9'd266 && screen_Y == 8'd232) ||
    (screen_X == 9'd266 && screen_Y == 8'd233) ||
    (screen_X == 9'd266 && screen_Y == 8'd234) ||
    (screen_X == 9'd266 && screen_Y == 8'd235) ||
    (screen_X == 9'd266 && screen_Y == 8'd236) ||
    (screen_X == 9'd271 && screen_Y == 8'd224) ||
    (screen_X == 9'd271 && screen_Y == 8'd225) ||
    (screen_X == 9'd271 && screen_Y == 8'd226) ||
    (screen_X == 9'd271 && screen_Y == 8'd227) ||
    (screen_X == 9'd271 && screen_Y == 8'd228) ||
    (screen_X == 9'd271 && screen_Y == 8'd229) ||
    (screen_X == 9'd271 && screen_Y == 8'd230) ||
    (screen_X == 9'd267 && screen_Y == 8'd224) ||
    (screen_X == 9'd268 && screen_Y == 8'd224) ||
    (screen_X == 9'd269 && screen_Y == 8'd224) ||

```

```

(screen_X == 9'd270 && screen_Y == 8'd224) ||
(screen_X == 9'd267 && screen_Y == 8'd230) ||
(screen_X == 9'd268 && screen_Y == 8'd230) ||
(screen_X == 9'd269 && screen_Y == 8'd230) ||
(screen_X == 9'd270 && screen_Y == 8'd230)
) begin // P
    pixel_colour = colourful ? (((randNum_12b[3:1] ^
randNum_12b[2:0]) == 3'b0) ? 3'b111 : (randNum_12b[3:1] ^ randNum_12b[2:0])) :
3'b111;

end

if ((screen_X == 9'd273 && screen_Y == 8'd226) ||
    (screen_X == 9'd273 && screen_Y == 8'd227) ||
    (screen_X == 9'd273 && screen_Y == 8'd228) ||
    (screen_X == 9'd273 && screen_Y == 8'd229) ||
    (screen_X == 9'd273 && screen_Y == 8'd230) ||
    (screen_X == 9'd273 && screen_Y == 8'd231) ||
    (screen_X == 9'd273 && screen_Y == 8'd232) ||
    (screen_X == 9'd273 && screen_Y == 8'd233) ||
    (screen_X == 9'd273 && screen_Y == 8'd234) ||
    (screen_X == 9'd273 && screen_Y == 8'd235) ||
    (screen_X == 9'd273 && screen_Y == 8'd236) ||
    (screen_X == 9'd273 && screen_Y == 8'd237) ||
    (screen_X == 9'd273 && screen_Y == 8'd238) ||
    (screen_X == 9'd274 && screen_Y == 8'd238) ||
    (screen_X == 9'd275 && screen_Y == 8'd238) ||
    (screen_X == 9'd276 && screen_Y == 8'd238) ||
    (screen_X == 9'd277 && screen_Y == 8'd238) ||
    (screen_X == 9'd278 && screen_Y == 8'd238) ||
    (screen_X == 9'd279 && screen_Y == 8'd238) ||

```

```

        (screen_X == 9'd280 && screen_Y == 8'd238)
    ) begin // I
        pixel_colour = colourful ? (((randNum_12b[3:1] ^
randNum_12b[3:1]) == 3'b0) ? 3'b111 : (randNum_12b[3:1] ^ randNum_12b[3:1])) :
3'b111;

    end

    if ((screen_X == 9'd279 && screen_Y == 8'd225) ||
        (screen_X == 9'd279 && screen_Y == 8'd226) ||
        (screen_X == 9'd279 && screen_Y == 8'd227) ||
        (screen_X == 9'd279 && screen_Y == 8'd228) ||
        (screen_X == 9'd279 && screen_Y == 8'd229) ||
        (screen_X == 9'd279 && screen_Y == 8'd230) ||
        (screen_X == 9'd279 && screen_Y == 8'd231) ||
        (screen_X == 9'd279 && screen_Y == 8'd232) ||
        (screen_X == 9'd279 && screen_Y == 8'd233) ||
        (screen_X == 9'd279 && screen_Y == 8'd234) ||
        (screen_X == 9'd279 && screen_Y == 8'd235) ||
        (screen_X == 9'd285 && screen_Y == 8'd225) ||
        (screen_X == 9'd285 && screen_Y == 8'd226) ||
        (screen_X == 9'd285 && screen_Y == 8'd227) ||
        (screen_X == 9'd285 && screen_Y == 8'd228) ||
        (screen_X == 9'd285 && screen_Y == 8'd229) ||
        (screen_X == 9'd285 && screen_Y == 8'd230) ||
        (screen_X == 9'd285 && screen_Y == 8'd231) ||
        (screen_X == 9'd285 && screen_Y == 8'd232) ||
        (screen_X == 9'd285 && screen_Y == 8'd233) ||
        (screen_X == 9'd285 && screen_Y == 8'd234) ||
        (screen_X == 9'd285 && screen_Y == 8'd235) ||
        (screen_X == 9'd285 && screen_Y == 8'd236) ||

```



```

(screen_X == 9'd280 && screen_Y == 8'd225) ||
(screen_X == 9'd281 && screen_Y == 8'd225) ||
(screen_X == 9'd282 && screen_Y == 8'd225) ||
(screen_X == 9'd283 && screen_Y == 8'd225) ||
(screen_X == 9'd284 && screen_Y == 8'd225) ||
(screen_X == 9'd280 && screen_Y == 8'd230) ||
(screen_X == 9'd281 && screen_Y == 8'd230) ||
(screen_X == 9'd282 && screen_Y == 8'd230) ||
(screen_X == 9'd283 && screen_Y == 8'd230) ||
(screen_X == 9'd284 && screen_Y == 8'd230)
) begin // a
    pixel_colour = colourful ? (((randNum_12b[3:1] ^
randNum_12b[4:3]) == 3'b0) ? 3'b111 : (randNum_12b[3:1] ^ randNum_12b[4:3])) :
3'b111;
end

if ((screen_X == 9'd288 && screen_Y == 8'd223) ||
    (screen_X == 9'd288 && screen_Y == 8'd224) ||
    (screen_X == 9'd288 && screen_Y == 8'd225) ||
    (screen_X == 9'd288 && screen_Y == 8'd226) ||
    (screen_X == 9'd288 && screen_Y == 8'd227) ||
    (screen_X == 9'd288 && screen_Y == 8'd228) ||
    (screen_X == 9'd288 && screen_Y == 8'd229) ||
    (screen_X == 9'd288 && screen_Y == 8'd230) ||
    (screen_X == 9'd288 && screen_Y == 8'd231) ||
    (screen_X == 9'd288 && screen_Y == 8'd232) ||
    (screen_X == 9'd288 && screen_Y == 8'd233) ||
    (screen_X == 9'd288 && screen_Y == 8'd234) ||
    (screen_X == 9'd288 && screen_Y == 8'd235) ||
    (screen_X == 9'd288 && screen_Y == 8'd236) ||

```

```

(screen_X == 9'd289 && screen_Y == 8'd223) ||
(screen_X == 9'd290 && screen_Y == 8'd223) ||
(screen_X == 9'd291 && screen_Y == 8'd223) ||
(screen_X == 9'd292 && screen_Y == 8'd223) ||
(screen_X == 9'd293 && screen_Y == 8'd223) ||
(screen_X == 9'd289 && screen_Y == 8'd229) ||
(screen_X == 9'd290 && screen_Y == 8'd229) ||
(screen_X == 9'd291 && screen_Y == 8'd229) ||
(screen_X == 9'd292 && screen_Y == 8'd229) ||
(screen_X == 9'd289 && screen_Y == 8'd235) ||
(screen_X == 9'd290 && screen_Y == 8'd235) ||
(screen_X == 9'd291 && screen_Y == 8'd235) ||
(screen_X == 9'd292 && screen_Y == 8'd235) ||
(screen_X == 9'd293 && screen_Y == 8'd235) ||
(screen_X == 9'd294 && screen_Y == 8'd235) ||
(screen_X == 9'd295 && screen_Y == 8'd235) ||
(screen_X == 9'd296 && screen_Y == 8'd235)
) begin // e

    pixel_colour = colourful ? (((randNum_12b[3:1] ^
randNum_12b[7:5]) == 3'b0) ? 3'b111 : (randNum_12b[3:1] ^ randNum_12b[7:5])) :
3'b111;

    end

    end

    end

    if (sig_drawBoard) begin

        if (screen_X >= 9'd240 && screen_X <= 9'd319 && screen_Y <=
8'd119) begin

            if ((screen_X == 9'd247 && screen_Y == 8'd8) ||

                (screen_X == 9'd247 && screen_Y == 8'd9) ||

```

(screen\_X == 9'd247 && screen\_Y == 8'd10) ||  
(screen\_X == 9'd247 && screen\_Y == 8'd11) ||  
(screen\_X == 9'd247 && screen\_Y == 8'd12) ||  
(screen\_X == 9'd247 && screen\_Y == 8'd13) ||  
(screen\_X == 9'd247 && screen\_Y == 8'd14) ||  
(screen\_X == 9'd247 && screen\_Y == 8'd15) ||  
(screen\_X == 9'd247 && screen\_Y == 8'd16) ||  
(screen\_X == 9'd247 && screen\_Y == 8'd17) ||  
(screen\_X == 9'd247 && screen\_Y == 8'd18) ||  
(screen\_X == 9'd247 && screen\_Y == 8'd19) ||  
(screen\_X == 9'd247 && screen\_Y == 8'd20) ||  
(screen\_X == 9'd248 && screen\_Y == 8'd13) ||  
(screen\_X == 9'd249 && screen\_Y == 8'd13) ||  
(screen\_X == 9'd250 && screen\_Y == 8'd13) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd13) ||  
(screen\_X == 9'd252 && screen\_Y == 8'd13) ||  
(screen\_X == 9'd253 && screen\_Y == 8'd13) ||  
(screen\_X == 9'd254 && screen\_Y == 8'd13) ||  
(screen\_X == 9'd255 && screen\_Y == 8'd8) ||  
(screen\_X == 9'd255 && screen\_Y == 8'd9) ||  
(screen\_X == 9'd255 && screen\_Y == 8'd10) ||  
(screen\_X == 9'd255 && screen\_Y == 8'd11) ||  
(screen\_X == 9'd255 && screen\_Y == 8'd12) ||  
(screen\_X == 9'd255 && screen\_Y == 8'd13) ||  
(screen\_X == 9'd255 && screen\_Y == 8'd14) ||  
(screen\_X == 9'd255 && screen\_Y == 8'd15) ||  
(screen\_X == 9'd255 && screen\_Y == 8'd16) ||  
(screen\_X == 9'd255 && screen\_Y == 8'd17) ||

```

(screen_X == 9'd255 && screen_Y == 8'd18) ||
(screen_X == 9'd255 && screen_Y == 8'd19) ||
(screen_X == 9'd255 && screen_Y == 8'd20)
) begin // H
    pixel_colour = colourful ? ((randNum_12b[11:9] ==
3'b0) ? 3'b111 : randNum_12b[11:9]) : 3'b111;
end
if ((screen_X == 9'd258 && screen_Y == 8'd9) ||
    (screen_X == 9'd258 && screen_Y == 8'd12) ||
    (screen_X == 9'd258 && screen_Y == 8'd13) ||
    (screen_X == 9'd258 && screen_Y == 8'd14) ||
    (screen_X == 9'd258 && screen_Y == 8'd15) ||
    (screen_X == 9'd258 && screen_Y == 8'd16) ||
    (screen_X == 9'd258 && screen_Y == 8'd17) ||
    (screen_X == 9'd258 && screen_Y == 8'd18)
) begin // i
    pixel_colour = colourful ? ((randNum_12b[10:8] ==
3'b0) ? 3'b111 : randNum_12b[10:8]) : 3'b111;
end
if ((screen_X == 9'd261 && screen_Y == 8'd12) ||
    (screen_X == 9'd261 && screen_Y == 8'd13) ||
    (screen_X == 9'd261 && screen_Y == 8'd14) ||
    (screen_X == 9'd261 && screen_Y == 8'd15) ||
    (screen_X == 9'd261 && screen_Y == 8'd16) ||
    (screen_X == 9'd261 && screen_Y == 8'd17) ||
    (screen_X == 9'd261 && screen_Y == 8'd18) ||
    (screen_X == 9'd265 && screen_Y == 8'd12) ||
    (screen_X == 9'd265 && screen_Y == 8'd13) ||
    (screen_X == 9'd265 && screen_Y == 8'd14) ||

```

```

(screen_X == 9'd265 && screen_Y == 8'd15) ||
(screen_X == 9'd265 && screen_Y == 8'd16) ||
(screen_X == 9'd265 && screen_Y == 8'd17) ||
(screen_X == 9'd265 && screen_Y == 8'd18) ||
(screen_X == 9'd265 && screen_Y == 8'd19) ||
(screen_X == 9'd265 && screen_Y == 8'd20) ||
(screen_X == 9'd265 && screen_Y == 8'd21) ||
(screen_X == 9'd265 && screen_Y == 8'd22) ||
(screen_X == 9'd265 && screen_Y == 8'd23) ||
(screen_X == 9'd265 && screen_Y == 8'd24) ||
(screen_X == 9'd262 && screen_Y == 8'd12) ||
(screen_X == 9'd263 && screen_Y == 8'd12) ||
(screen_X == 9'd264 && screen_Y == 8'd12) ||
(screen_X == 9'd262 && screen_Y == 8'd18) ||
(screen_X == 9'd263 && screen_Y == 8'd18) ||
(screen_X == 9'd264 && screen_Y == 8'd18) ||
(screen_X == 9'd261 && screen_Y == 8'd22) ||
(screen_X == 9'd261 && screen_Y == 8'd23) ||
(screen_X == 9'd261 && screen_Y == 8'd24) ||
(screen_X == 9'd262 && screen_Y == 8'd24) ||
(screen_X == 9'd263 && screen_Y == 8'd24) ||
(screen_X == 9'd264 && screen_Y == 8'd24)
) begin // g
    pixel_colour = colourful ? ((randNum_12b[9:7] ==
3'b0) ? 3'b111 : randNum_12b[9:7]) : 3'b111;
end
if ((screen_X == 9'd267 && screen_Y == 8'd8) ||
    (screen_X == 9'd268 && screen_Y == 8'd8) ||
    (screen_X == 9'd268 && screen_Y == 8'd9) ||

```

```

(screen_X == 9'd268 && screen_Y == 8'd10) ||
(screen_X == 9'd268 && screen_Y == 8'd11) ||
(screen_X == 9'd268 && screen_Y == 8'd12) ||
(screen_X == 9'd268 && screen_Y == 8'd13) ||
(screen_X == 9'd268 && screen_Y == 8'd14) ||
(screen_X == 9'd268 && screen_Y == 8'd15) ||
(screen_X == 9'd268 && screen_Y == 8'd16) ||
(screen_X == 9'd268 && screen_Y == 8'd17) ||
(screen_X == 9'd268 && screen_Y == 8'd18) ||
(screen_X == 9'd268 && screen_Y == 8'd19) ||
(screen_X == 9'd269 && screen_Y == 8'd13) ||
(screen_X == 9'd270 && screen_Y == 8'd13) ||
(screen_X == 9'd271 && screen_Y == 8'd13) ||
(screen_X == 9'd272 && screen_Y == 8'd13) ||
(screen_X == 9'd272 && screen_Y == 8'd14) ||
(screen_X == 9'd272 && screen_Y == 8'd15) ||
(screen_X == 9'd272 && screen_Y == 8'd16) ||
(screen_X == 9'd272 && screen_Y == 8'd17) ||
(screen_X == 9'd272 && screen_Y == 8'd18)
) begin // h
    pixel_colour = colourful ? ((randNum_12b[8:6] ==
3'b0) ? 3'b111 : randNum_12b[8:6]) : 3'b111;
end
if ((screen_X == 9'd276 && screen_Y == 8'd12) ||
    (screen_X == 9'd277 && screen_Y == 8'd12) ||
    (screen_X == 9'd278 && screen_Y == 8'd12) ||
    (screen_X == 9'd279 && screen_Y == 8'd12) ||
    (screen_X == 9'd280 && screen_Y == 8'd12) ||
    (screen_X == 9'd281 && screen_Y == 8'd12) ||

```

```

(screen_X == 9'd276 && screen_Y == 8'd13) ||
(screen_X == 9'd276 && screen_Y == 8'd14) ||
(screen_X == 9'd276 && screen_Y == 8'd15) ||
(screen_X == 9'd276 && screen_Y == 8'd16) ||
(screen_X == 9'd276 && screen_Y == 8'd17) ||
(screen_X == 9'd276 && screen_Y == 8'd18) ||
(screen_X == 9'd276 && screen_Y == 8'd19) ||
(screen_X == 9'd277 && screen_Y == 8'd15) ||
(screen_X == 9'd278 && screen_Y == 8'd15) ||
(screen_X == 9'd279 && screen_Y == 8'd15) ||
(screen_X == 9'd280 && screen_Y == 8'd15) ||
(screen_X == 9'd281 && screen_Y == 8'd15) ||
(screen_X == 9'd282 && screen_Y == 8'd15) ||
(screen_X == 9'd277 && screen_Y == 8'd19) ||
(screen_X == 9'd278 && screen_Y == 8'd19) ||
(screen_X == 9'd279 && screen_Y == 8'd19) ||
(screen_X == 9'd280 && screen_Y == 8'd19) ||
(screen_X == 9'd281 && screen_Y == 8'd19) ||
(screen_X == 9'd282 && screen_Y == 8'd19) ||
(screen_X == 9'd281 && screen_Y == 8'd13) ||
(screen_X == 9'd282 && screen_Y == 8'd13) ||
(screen_X == 9'd282 && screen_Y == 8'd14)
) begin // e
    pixel_colour = colourful ? ((randNum_12b[7:5] ==
3'b0) ? 3'b111 : randNum_12b[7:5]) : 3'b111;
end
if ((screen_X == 9'd286 && screen_Y == 8'd13) ||
    (screen_X == 9'd287 && screen_Y == 8'd13) ||
    (screen_X == 9'd288 && screen_Y == 8'd13) ||

```

```

(screen_X == 9'd289 && screen_Y == 8'd13) ||
(screen_X == 9'd290 && screen_Y == 8'd13) ||
(screen_X == 9'd291 && screen_Y == 8'd13) ||
(screen_X == 9'd292 && screen_Y == 8'd13) ||
(screen_X == 9'd293 && screen_Y == 8'd13) ||
(screen_X == 9'd286 && screen_Y == 8'd14) ||
(screen_X == 9'd286 && screen_Y == 8'd15) ||
(screen_X == 9'd286 && screen_Y == 8'd16) ||
(screen_X == 9'd286 && screen_Y == 8'd17) ||
(screen_X == 9'd287 && screen_Y == 8'd17) ||
(screen_X == 9'd288 && screen_Y == 8'd17) ||
(screen_X == 9'd289 && screen_Y == 8'd17) ||
(screen_X == 9'd290 && screen_Y == 8'd17) ||
(screen_X == 9'd291 && screen_Y == 8'd17) ||
(screen_X == 9'd292 && screen_Y == 8'd17) ||
(screen_X == 9'd293 && screen_Y == 8'd17) ||
(screen_X == 9'd293 && screen_Y == 8'd18) ||
(screen_X == 9'd293 && screen_Y == 8'd19) ||
(screen_X == 9'd287 && screen_Y == 8'd20) ||
(screen_X == 9'd288 && screen_Y == 8'd20) ||
(screen_X == 9'd289 && screen_Y == 8'd20) ||
(screen_X == 9'd290 && screen_Y == 8'd20) ||
(screen_X == 9'd291 && screen_Y == 8'd20) ||
(screen_X == 9'd292 && screen_Y == 8'd20) ||
(screen_X == 9'd293 && screen_Y == 8'd20)
) begin // s
    pixel_colour = colourful ? ((randNum_12b[6:4] ==
3'b0) ? 3'b111 : randNum_12b[6:4]) : 3'b111;
end

```



```

if ((screen_X == 9'd295 && screen_Y == 8'd10) ||
    (screen_X == 9'd296 && screen_Y == 8'd10) ||
    (screen_X == 9'd297 && screen_Y == 8'd10) ||
    (screen_X == 9'd298 && screen_Y == 8'd10) ||
    (screen_X == 9'd299 && screen_Y == 8'd10) ||
    (screen_X == 9'd300 && screen_Y == 8'd10) ||
    (screen_X == 9'd301 && screen_Y == 8'd10) ||
    (screen_X == 9'd302 && screen_Y == 8'd10) ||
    (screen_X == 9'd303 && screen_Y == 8'd10) ||
    (screen_X == 9'd304 && screen_Y == 8'd10) ||
    (screen_X == 9'd305 && screen_Y == 8'd10) ||
    (screen_X == 9'd300 && screen_Y == 8'd11) ||
    (screen_X == 9'd300 && screen_Y == 8'd12) ||
    (screen_X == 9'd300 && screen_Y == 8'd13) ||
    (screen_X == 9'd300 && screen_Y == 8'd14) ||
    (screen_X == 9'd300 && screen_Y == 8'd15) ||
    (screen_X == 9'd300 && screen_Y == 8'd16) ||
    (screen_X == 9'd300 && screen_Y == 8'd17) ||
    (screen_X == 9'd300 && screen_Y == 8'd18) ||
    (screen_X == 9'd300 && screen_Y == 8'd19) ||
    (screen_X == 9'd300 && screen_Y == 8'd20) ||
    (screen_X == 9'd300 && screen_Y == 8'd21)
) begin // t
    pixel_colour = colourful ? ((randNum_12b[5:3] ==
3'b0) ? 3'b111 : randNum_12b[5:3]) : 3'b111;
end

if ((screen_X == 9'd258 && screen_Y == 8'd29) ||
    (screen_X == 9'd256 && screen_Y == 8'd30) ||
    (screen_X == 9'd257 && screen_Y == 8'd30) ||

```

```

(screen_X == 9'd255 && screen_Y == 8'd31) ||
(screen_X == 9'd255 && screen_Y == 8'd32) ||
(screen_X == 9'd255 && screen_Y == 8'd33) ||
(screen_X == 9'd255 && screen_Y == 8'd34) ||
(screen_X == 9'd256 && screen_Y == 8'd35) ||
(screen_X == 9'd256 && screen_Y == 8'd36) ||
(screen_X == 9'd256 && screen_Y == 8'd37) ||
(screen_X == 9'd257 && screen_Y == 8'd38) ||
(screen_X == 9'd257 && screen_Y == 8'd39) ||
(screen_X == 9'd258 && screen_Y == 8'd40) ||
(screen_X == 9'd258 && screen_Y == 8'd41) ||
(screen_X == 9'd259 && screen_Y == 8'd41) ||
(screen_X == 9'd259 && screen_Y == 8'd42) ||
(screen_X == 9'd260 && screen_Y == 8'd42) ||
(screen_X == 9'd254 && screen_Y == 8'd42) ||
(screen_X == 9'd255 && screen_Y == 8'd42) ||
(screen_X == 9'd256 && screen_Y == 8'd42) ||
(screen_X == 9'd257 && screen_Y == 8'd43) ||
(screen_X == 9'd258 && screen_Y == 8'd43) ||
(screen_X == 9'd259 && screen_Y == 8'd43) ||
(screen_X == 9'd260 && screen_Y == 8'd43) ||
(screen_X == 9'd261 && screen_Y == 8'd43) ||
(screen_X == 9'd261 && screen_Y == 8'd38) ||
(screen_X == 9'd261 && screen_Y == 8'd39) ||
(screen_X == 9'd261 && screen_Y == 8'd40) ||
(screen_X == 9'd261 && screen_Y == 8'd41) ||
(screen_X == 9'd261 && screen_Y == 8'd42)
) begin // arrow

```

```
        pixel_colour = colourful ? ((randNum_12b[4:3] ==  
3'b0) ? 3'b111 : randNum_12b[4:3]) : 3'b111;
```

```
    end
```

```
    if ((screen_X == 9'd273 && screen_Y == 8'd29) ||
```

```
        (screen_X == 9'd274 && screen_Y == 8'd29) ||
```

```
        (screen_X == 9'd275 && screen_Y == 8'd29) ||
```

```
        (screen_X == 9'd276 && screen_Y == 8'd29) ||
```

```
        (screen_X == 9'd277 && screen_Y == 8'd29) ||
```

```
        (screen_X == 9'd278 && screen_Y == 8'd29) ||
```

```
        (screen_X == 9'd279 && screen_Y == 8'd29) ||
```

```
        (screen_X == 9'd280 && screen_Y == 8'd29) ||
```

```
        (screen_X == 9'd281 && screen_Y == 8'd29) ||
```

```
        (screen_X == 9'd281 && screen_Y == 8'd30) ||
```

```
        (screen_X == 9'd281 && screen_Y == 8'd31) ||
```

```
        (screen_X == 9'd281 && screen_Y == 8'd36) ||
```

```
        (screen_X == 9'd281 && screen_Y == 8'd37) ||
```

```
        (screen_X == 9'd281 && screen_Y == 8'd38) ||
```

```
        (screen_X == 9'd281 && screen_Y == 8'd39) ||
```

```
        (screen_X == 9'd281 && screen_Y == 8'd40) ||
```

```
        (screen_X == 9'd281 && screen_Y == 8'd41) ||
```

```
        (screen_X == 9'd273 && screen_Y == 8'd30) ||
```

```
        (screen_X == 9'd273 && screen_Y == 8'd31) ||
```

```
        (screen_X == 9'd273 && screen_Y == 8'd32) ||
```

```
        (screen_X == 9'd273 && screen_Y == 8'd33) ||
```

```
        (screen_X == 9'd273 && screen_Y == 8'd34) ||
```

```
        (screen_X == 9'd273 && screen_Y == 8'd35) ||
```

```
        (screen_X == 9'd274 && screen_Y == 8'd35) ||
```

```
        (screen_X == 9'd275 && screen_Y == 8'd35) ||
```

```
        (screen_X == 9'd276 && screen_Y == 8'd35) ||
```

```

(screen_X == 9'd277 && screen_Y == 8'd35) ||
(screen_X == 9'd278 && screen_Y == 8'd35) ||
(screen_X == 9'd279 && screen_Y == 8'd35) ||
(screen_X == 9'd280 && screen_Y == 8'd35) ||
(screen_X == 9'd281 && screen_Y == 8'd35) ||
(screen_X == 9'd272 && screen_Y == 8'd39) ||
(screen_X == 9'd272 && screen_Y == 8'd40) ||
(screen_X == 9'd272 && screen_Y == 8'd41) ||
(screen_X == 9'd273 && screen_Y == 8'd41) ||
(screen_X == 9'd274 && screen_Y == 8'd41) ||
(screen_X == 9'd275 && screen_Y == 8'd41) ||
(screen_X == 9'd276 && screen_Y == 8'd41) ||
(screen_X == 9'd277 && screen_Y == 8'd41) ||
(screen_X == 9'd278 && screen_Y == 8'd41) ||
(screen_X == 9'd279 && screen_Y == 8'd41) ||
(screen_X == 9'd280 && screen_Y == 8'd41)
) begin // S
    pixel_colour = colourful ? ((randNum_12b[3:1] ==
3'b0) ? 3'b111 : randNum_12b[3:1]) : 3'b111;
end
if ((screen_X == 9'd283 && screen_Y == 8'd32) ||
(screen_X == 9'd284 && screen_Y == 8'd32) ||
(screen_X == 9'd285 && screen_Y == 8'd32) ||
(screen_X == 9'd286 && screen_Y == 8'd32) ||
(screen_X == 9'd287 && screen_Y == 8'd32) ||
(screen_X == 9'd288 && screen_Y == 8'd32) ||
(screen_X == 9'd288 && screen_Y == 8'd33) ||
(screen_X == 9'd283 && screen_Y == 8'd33) ||
(screen_X == 9'd283 && screen_Y == 8'd34) ||

```

```

(screen_X == 9'd283 && screen_Y == 8'd35) ||
(screen_X == 9'd283 && screen_Y == 8'd36) ||
(screen_X == 9'd283 && screen_Y == 8'd37) ||
(screen_X == 9'd283 && screen_Y == 8'd38) ||
(screen_X == 9'd283 && screen_Y == 8'd39) ||
(screen_X == 9'd284 && screen_Y == 8'd39) ||
(screen_X == 9'd285 && screen_Y == 8'd39) ||
(screen_X == 9'd286 && screen_Y == 8'd39) ||
(screen_X == 9'd287 && screen_Y == 8'd39) ||
(screen_X == 9'd288 && screen_Y == 8'd39)
) begin // c
    pixel_colour = colourful ? ((randNum_12b[2:0] ==
3'b0) ? 3'b111 : randNum_12b[2:0]) : 3'b111;
end
if ((screen_X == 9'd291 && screen_Y == 8'd32) ||
(screen_X == 9'd292 && screen_Y == 8'd32) ||
(screen_X == 9'd293 && screen_Y == 8'd32) ||
(screen_X == 9'd294 && screen_Y == 8'd32) ||
(screen_X == 9'd295 && screen_Y == 8'd32) ||
(screen_X == 9'd296 && screen_Y == 8'd32) ||
(screen_X == 9'd297 && screen_Y == 8'd32) ||
(screen_X == 9'd292 && screen_Y == 8'd39) ||
(screen_X == 9'd293 && screen_Y == 8'd39) ||
(screen_X == 9'd294 && screen_Y == 8'd39) ||
(screen_X == 9'd295 && screen_Y == 8'd39) ||
(screen_X == 9'd296 && screen_Y == 8'd39) ||
(screen_X == 9'd291 && screen_Y == 8'd33) ||
(screen_X == 9'd291 && screen_Y == 8'd34) ||
(screen_X == 9'd291 && screen_Y == 8'd35) ||

```

```

(screen_X == 9'd291 && screen_Y == 8'd36) ||
(screen_X == 9'd291 && screen_Y == 8'd37) ||
(screen_X == 9'd291 && screen_Y == 8'd38) ||
(screen_X == 9'd291 && screen_Y == 8'd39) ||
(screen_X == 9'd297 && screen_Y == 8'd33) ||
(screen_X == 9'd297 && screen_Y == 8'd34) ||
(screen_X == 9'd297 && screen_Y == 8'd35) ||
(screen_X == 9'd297 && screen_Y == 8'd36) ||
(screen_X == 9'd297 && screen_Y == 8'd37) ||
(screen_X == 9'd297 && screen_Y == 8'd38) ||
(screen_X == 9'd297 && screen_Y == 8'd39) ||
(screen_X == 9'd297 && screen_Y == 8'd40)
) begin // o

    pixel_colour = colourful ? (((randNum_12b[11:9] ^
randNum_12b[2:0]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^ randNum_12b[2:0])) :
3'b111;

end

if ((screen_X == 9'd300 && screen_Y == 8'd33) ||
    (screen_X == 9'd300 && screen_Y == 8'd34) ||
    (screen_X == 9'd300 && screen_Y == 8'd35) ||
    (screen_X == 9'd300 && screen_Y == 8'd36) ||
    (screen_X == 9'd300 && screen_Y == 8'd37) ||
    (screen_X == 9'd300 && screen_Y == 8'd38) ||
    (screen_X == 9'd300 && screen_Y == 8'd39) ||
    (screen_X == 9'd301 && screen_Y == 8'd34) ||
    (screen_X == 9'd302 && screen_Y == 8'd33) ||
    (screen_X == 9'd303 && screen_Y == 8'd33) ||
    (screen_X == 9'd304 && screen_Y == 8'd33) ||
    (screen_X == 9'd305 && screen_Y == 8'd33) ||

```

```

(screen_X == 9'd306 && screen_Y == 8'd33)
) begin // r
    pixel_colour = colourful ? (((randNum_12b[10:8] ^
randNum_12b[2:0]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^ randNum_12b[2:0])) :
3'b111;

end

if ((screen_X == 9'd310 && screen_Y == 8'd32) ||
    (screen_X == 9'd311 && screen_Y == 8'd32) ||
    (screen_X == 9'd312 && screen_Y == 8'd32) ||
    (screen_X == 9'd313 && screen_Y == 8'd32) ||
    (screen_X == 9'd314 && screen_Y == 8'd33) ||
    (screen_X == 9'd314 && screen_Y == 8'd34) ||
    (screen_X == 9'd315 && screen_Y == 8'd34) ||
    (screen_X == 9'd315 && screen_Y == 8'd35) ||
    (screen_X == 9'd315 && screen_Y == 8'd36) ||
    (screen_X == 9'd310 && screen_Y == 8'd37) ||
    (screen_X == 9'd311 && screen_Y == 8'd37) ||
    (screen_X == 9'd312 && screen_Y == 8'd37) ||
    (screen_X == 9'd313 && screen_Y == 8'd37) ||
    (screen_X == 9'd314 && screen_Y == 8'd37) ||
    (screen_X == 9'd315 && screen_Y == 8'd37) ||
    (screen_X == 9'd311 && screen_Y == 8'd41) ||
    (screen_X == 9'd312 && screen_Y == 8'd41) ||
    (screen_X == 9'd313 && screen_Y == 8'd41) ||
    (screen_X == 9'd314 && screen_Y == 8'd41) ||
    (screen_X == 9'd315 && screen_Y == 8'd41) ||
    (screen_X == 9'd310 && screen_Y == 8'd40) ||
    (screen_X == 9'd309 && screen_Y == 8'd33) ||
    (screen_X == 9'd309 && screen_Y == 8'd34) ||

```

```

(screen_X == 9'd309 && screen_Y == 8'd35) ||
(screen_X == 9'd309 && screen_Y == 8'd36) ||
(screen_X == 9'd309 && screen_Y == 8'd37) ||
(screen_X == 9'd309 && screen_Y == 8'd38) ||
(screen_X == 9'd309 && screen_Y == 8'd39)
) begin // e
    pixel_colour = colourful ? (((randNum_12b[11:9] ^
randNum_12b[3:1]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^ randNum_12b[3:1])) :
3'b111;

end

if ((screen_X == 9'd251 && screen_Y == 8'd50) ||
    (screen_X == 9'd252 && screen_Y == 8'd50) ||
    (screen_X == 9'd253 && screen_Y == 8'd50) ||
    (screen_X == 9'd254 && screen_Y == 8'd50) ||
    (screen_X == 9'd255 && screen_Y == 8'd50) ||
    (screen_X == 9'd256 && screen_Y == 8'd50) ||
    (screen_X == 9'd257 && screen_Y == 8'd50) ||
    (screen_X == 9'd258 && screen_Y == 8'd50) ||
    (screen_X == 9'd259 && screen_Y == 8'd50) ||
    (screen_X == 9'd260 && screen_Y == 8'd50) ||
    (screen_X == 9'd261 && screen_Y == 8'd50) ||
    (screen_X == 9'd262 && screen_Y == 8'd50) ||
    (screen_X == 9'd263 && screen_Y == 8'd50) ||
    (screen_X == 9'd264 && screen_Y == 8'd50) ||
    (screen_X == 9'd265 && screen_Y == 8'd50) ||
    (screen_X == 9'd266 && screen_Y == 8'd50) ||
    (screen_X == 9'd267 && screen_Y == 8'd50) ||
    (screen_X == 9'd268 && screen_Y == 8'd50) ||
    (screen_X == 9'd269 && screen_Y == 8'd50) ||

```



(screen\_X == 9'd270 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd271 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd272 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd273 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd274 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd275 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd276 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd277 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd278 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd279 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd280 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd281 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd282 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd283 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd284 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd285 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd286 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd287 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd288 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd289 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd290 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd291 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd292 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd293 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd294 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd295 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd296 && screen\_Y == 8'd50) ||  
(screen\_X == 9'd297 && screen\_Y == 8'd50) ||

```
(screen_X == 9'd298 && screen_Y == 8'd50) ||  
(screen_X == 9'd299 && screen_Y == 8'd50) ||  
(screen_X == 9'd300 && screen_Y == 8'd50) ||  
(screen_X == 9'd301 && screen_Y == 8'd50) ||  
(screen_X == 9'd302 && screen_Y == 8'd50) ||  
(screen_X == 9'd303 && screen_Y == 8'd50) ||  
(screen_X == 9'd304 && screen_Y == 8'd50) ||  
(screen_X == 9'd305 && screen_Y == 8'd50) ||  
(screen_X == 9'd306 && screen_Y == 8'd50) ||  
(screen_X == 9'd307 && screen_Y == 8'd50) ||  
(screen_X == 9'd308 && screen_Y == 8'd50) ||  
(screen_X == 9'd309 && screen_Y == 8'd50) ||  
(screen_X == 9'd251 && screen_Y == 8'd108) ||  
(screen_X == 9'd252 && screen_Y == 8'd108) ||  
(screen_X == 9'd253 && screen_Y == 8'd108) ||  
(screen_X == 9'd254 && screen_Y == 8'd108) ||  
(screen_X == 9'd255 && screen_Y == 8'd108) ||  
(screen_X == 9'd256 && screen_Y == 8'd108) ||  
(screen_X == 9'd257 && screen_Y == 8'd108) ||  
(screen_X == 9'd258 && screen_Y == 8'd108) ||  
(screen_X == 9'd259 && screen_Y == 8'd108) ||  
(screen_X == 9'd260 && screen_Y == 8'd108) ||  
(screen_X == 9'd261 && screen_Y == 8'd108) ||  
(screen_X == 9'd262 && screen_Y == 8'd108) ||  
(screen_X == 9'd263 && screen_Y == 8'd108) ||  
(screen_X == 9'd264 && screen_Y == 8'd108) ||  
(screen_X == 9'd265 && screen_Y == 8'd108) ||  
(screen_X == 9'd266 && screen_Y == 8'd108) ||
```

(screen\_X == 9'd267 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd268 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd269 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd270 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd271 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd272 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd273 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd274 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd275 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd276 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd277 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd278 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd279 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd280 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd281 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd282 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd283 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd284 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd285 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd286 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd287 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd288 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd289 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd290 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd291 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd292 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd293 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd294 && screen\_Y == 8'd108) ||

(screen\_X == 9'd295 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd296 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd297 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd298 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd299 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd300 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd301 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd302 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd303 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd304 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd305 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd306 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd307 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd308 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd108) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd51) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd52) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd53) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd54) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd56) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd57) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd58) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd59) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd60) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd61) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd62) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd63) ||

(screen\_X == 9'd251 && screen\_Y == 8'd64) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd65) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd66) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd67) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd68) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd69) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd70) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd71) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd72) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd73) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd74) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd75) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd76) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd77) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd78) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd79) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd80) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd81) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd82) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd83) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd84) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd85) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd86) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd87) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd88) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd90) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd91) ||

(screen\_X == 9'd251 && screen\_Y == 8'd92) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd93) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd94) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd95) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd96) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd97) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd98) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd99) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd100) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd101) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd102) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd103) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd104) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd105) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd106) ||  
(screen\_X == 9'd251 && screen\_Y == 8'd107) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd51) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd52) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd53) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd54) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd55) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd56) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd57) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd58) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd59) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd60) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd61) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd62) ||

(screen\_X == 9'd309 && screen\_Y == 8'd63) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd64) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd65) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd66) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd67) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd68) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd69) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd70) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd71) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd72) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd73) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd74) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd75) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd76) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd77) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd78) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd79) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd80) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd81) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd82) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd83) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd84) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd85) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd86) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd87) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd88) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd89) ||  
(screen\_X == 9'd309 && screen\_Y == 8'd90) ||

```

(screen_X == 9'd309 && screen_Y == 8'd91) ||
(screen_X == 9'd309 && screen_Y == 8'd92) ||
(screen_X == 9'd309 && screen_Y == 8'd93) ||
(screen_X == 9'd309 && screen_Y == 8'd94) ||
(screen_X == 9'd309 && screen_Y == 8'd95) ||
(screen_X == 9'd309 && screen_Y == 8'd96) ||
(screen_X == 9'd309 && screen_Y == 8'd97) ||
(screen_X == 9'd309 && screen_Y == 8'd98) ||
(screen_X == 9'd309 && screen_Y == 8'd99) ||
(screen_X == 9'd309 && screen_Y == 8'd100) ||
(screen_X == 9'd309 && screen_Y == 8'd101) ||
(screen_X == 9'd309 && screen_Y == 8'd102) ||
(screen_X == 9'd309 && screen_Y == 8'd103) ||
(screen_X == 9'd309 && screen_Y == 8'd104) ||
(screen_X == 9'd309 && screen_Y == 8'd105) ||
(screen_X == 9'd309 && screen_Y == 8'd106) ||
(screen_X == 9'd309 && screen_Y == 8'd107)
) begin // box border

    pixel_colour = colourful ? (((randNum_12b[6:4] ^
randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[6:4] ^ randNum_12b[4:2])) :
3'b111;

    end

    if (screen_X >= 9'd252 && screen_X <= 9'd308 && screen_Y
>= 8'd50 && screen_Y <= 8'd106) begin
        effective_X = screen_X - 9'd252;
        effective_Y = screen_Y - 8'd51;
        if (highscore == 12'd0) begin
            end
        if (highscore == 12'd2) begin

```





```

                (effective_X == 9'd50 && effective_Y ==
8'd18) || (effective_X == 9'd50 && effective_Y == 8'd19) || (effective_X == 9'd50 &&
effective_Y == 8'd27) || (effective_X == 9'd50 && effective_Y == 8'd28) || (effective_X
== 9'd50 && effective_Y == 8'd37) || (effective_X == 9'd50 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd51 && effective_Y ==
8'd18) || (effective_X == 9'd51 && effective_Y == 8'd19) || (effective_X == 9'd51 &&
effective_Y == 8'd27) || (effective_X == 9'd51 && effective_Y == 8'd28) || (effective_X
== 9'd51 && effective_Y == 8'd37) || (effective_X == 9'd51 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd52 && effective_Y ==
8'd18) || (effective_X == 9'd52 && effective_Y == 8'd19) || (effective_X == 9'd52 &&
effective_Y == 8'd27) || (effective_X == 9'd52 && effective_Y == 8'd28) || (effective_X
== 9'd52 && effective_Y == 8'd37) || (effective_X == 9'd52 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd53 && effective_Y ==
8'd18) || (effective_X == 9'd53 && effective_Y == 8'd19) || (effective_X == 9'd53 &&
effective_Y == 8'd20) || (effective_X == 9'd53 && effective_Y == 8'd21) || (effective_X
== 9'd53 && effective_Y == 8'd22) || (effective_X == 9'd53 && effective_Y == 8'd23) ||
(effective_X == 9'd53 && effective_Y == 8'd24) || (effective_X == 9'd53 && effective_Y
== 8'd25) || (effective_X == 9'd53 && effective_Y == 8'd26) || (effective_X == 9'd53 &&
effective_Y == 8'd27) || (effective_X == 9'd53 && effective_Y == 8'd28) || (effective_X
== 9'd53 && effective_Y == 8'd37) || (effective_X == 9'd53 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd37) || (effective_X == 9'd54 && effective_Y == 8'd38)

```

```

            ) begin

```

```

                pixel_colour = colourful ?
(((randNum_12b[7:5] ^ randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[7:5] ^
randNum_12b[4:2])) : 3'b111;

```

```

            end

```

```

        end

```

```

        if (highscore == 12'd4) begin

```

```

                if((effective_X == 9'd43 && effective_Y ==
8'd18) || (effective_X == 9'd43 && effective_Y == 8'd19) || (effective_X == 9'd43 &&
effective_Y == 8'd20) || (effective_X == 9'd43 && effective_Y == 8'd21) || (effective_X

```



```

== 9'd53 && effective_Y == 8'd36) || (effective_X == 9'd53 && effective_Y == 8'd37) ||
(effective_X == 9'd53 && effective_Y == 8'd38) ||

(effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

) begin

pixel_colour = colourful ?
(((randNum_12b[10:8] ^ randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^
randNum_12b[4:2])) : 3'b111;

end

end

if (highscore == 12'd8) begin

if((effective_X == 9'd43 && effective_Y ==
8'd18) || (effective_X == 9'd43 && effective_Y == 8'd19) || (effective_X == 9'd43 &&
effective_Y == 8'd20) || (effective_X == 9'd43 && effective_Y == 8'd21) || (effective_X
== 9'd43 && effective_Y == 8'd22) || (effective_X == 9'd43 && effective_Y == 8'd23) ||
(effective_X == 9'd43 && effective_Y == 8'd24) || (effective_X == 9'd43 && effective_Y
== 8'd25) || (effective_X == 9'd43 && effective_Y == 8'd26) || (effective_X == 9'd43 &&
effective_Y == 8'd27) || (effective_X == 9'd43 && effective_Y == 8'd28) || (effective_X
== 9'd43 && effective_Y == 8'd29) || (effective_X == 9'd43 && effective_Y == 8'd30) ||
(effective_X == 9'd43 && effective_Y == 8'd31) || (effective_X == 9'd43 && effective_Y
== 8'd32) || (effective_X == 9'd43 && effective_Y == 8'd33) || (effective_X == 9'd43 &&
effective_Y == 8'd34) || (effective_X == 9'd43 && effective_Y == 8'd35) || (effective_X
== 9'd43 && effective_Y == 8'd36) || (effective_X == 9'd43 && effective_Y == 8'd37) ||
(effective_X == 9'd43 && effective_Y == 8'd38) ||

(effective_X == 9'd44 && effective_Y ==
8'd18) || (effective_X == 9'd44 && effective_Y == 8'd19) || (effective_X == 9'd44 &&
effective_Y == 8'd20) || (effective_X == 9'd44 && effective_Y == 8'd21) || (effective_X
== 9'd44 && effective_Y == 8'd22) || (effective_X == 9'd44 && effective_Y == 8'd23) ||

```



```

effective_Y == 8'd27) || (effective_X == 9'd52 && effective_Y == 8'd28) || (effective_X
== 9'd52 && effective_Y == 8'd37) || (effective_X == 9'd52 && effective_Y == 8'd38) ||

                                (effective_X == 9'd53 && effective_Y ==
8'd18) || (effective_X == 9'd53 && effective_Y == 8'd19) || (effective_X == 9'd53 &&
effective_Y == 8'd20) || (effective_X == 9'd53 && effective_Y == 8'd21) || (effective_X
== 9'd53 && effective_Y == 8'd22) || (effective_X == 9'd53 && effective_Y == 8'd23) ||
(effective_X == 9'd53 && effective_Y == 8'd24) || (effective_X == 9'd53 && effective_Y
== 8'd25) || (effective_X == 9'd53 && effective_Y == 8'd26) || (effective_X == 9'd53 &&
effective_Y == 8'd27) || (effective_X == 9'd53 && effective_Y == 8'd28) || (effective_X
== 9'd53 && effective_Y == 8'd29) || (effective_X == 9'd53 && effective_Y == 8'd30) ||
(effective_X == 9'd53 && effective_Y == 8'd31) || (effective_X == 9'd53 && effective_Y
== 8'd32) || (effective_X == 9'd53 && effective_Y == 8'd33) || (effective_X == 9'd53 &&
effective_Y == 8'd34) || (effective_X == 9'd53 && effective_Y == 8'd35) || (effective_X
== 9'd53 && effective_Y == 8'd36) || (effective_X == 9'd53 && effective_Y == 8'd37) ||
(effective_X == 9'd53 && effective_Y == 8'd38) ||

                                (effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

                                ) begin

                                pixel_colour = colourful ?

                                (((randNum_12b[10:8] ^ randNum_12b[3:1]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^
randNum_12b[3:1])) : 3'b111;

                                end

                                end

                                if (highscore == 12'd16) begin

                                if((effective_X == 9'd39 && effective_Y ==
8'd18) || (effective_X == 9'd39 && effective_Y == 8'd19) || (effective_X == 9'd39 &&
effective_Y == 8'd20) || (effective_X == 9'd39 && effective_Y == 8'd21) || (effective_X
== 9'd39 && effective_Y == 8'd22) || (effective_X == 9'd39 && effective_Y == 8'd23) ||

```







```

                (effective_X == 9'd53 && effective_Y ==
8'd18) || (effective_X == 9'd53 && effective_Y == 8'd19) || (effective_X == 9'd53 &&
effective_Y == 8'd27) || (effective_X == 9'd53 && effective_Y == 8'd28) || (effective_X
== 9'd53 && effective_Y == 8'd29) || (effective_X == 9'd53 && effective_Y == 8'd30) ||
(effective_X == 9'd53 && effective_Y == 8'd31) || (effective_X == 9'd53 && effective_Y
== 8'd32) || (effective_X == 9'd53 && effective_Y == 8'd33) || (effective_X == 9'd53 &&
effective_Y == 8'd34) || (effective_X == 9'd53 && effective_Y == 8'd35) || (effective_X
== 9'd53 && effective_Y == 8'd36) || (effective_X == 9'd53 && effective_Y == 8'd37) ||
(effective_X == 9'd53 && effective_Y == 8'd38)||

```

```

                (effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

```

```

            ) begin

```

```

                pixel_colour = colourful ?

```

```

                (((randNum_12b[6:4] ^ randNum_12b[7:5]) == 3'b0) ? 3'b111 : (randNum_12b[6:4] ^
randNum_12b[7:5])) : 3'b111;

```

```

            end

```

```

        end

```

```

        if (highscore == 12'd32) begin

```

```

                if((effective_X == 9'd29 && effective_Y ==
8'd18) || (effective_X == 9'd29 && effective_Y == 8'd19) || (effective_X == 9'd29 &&
effective_Y == 8'd27) || (effective_X == 9'd29 && effective_Y == 8'd28) || (effective_X
== 9'd29 && effective_Y == 8'd37) || (effective_X == 9'd29 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd30 && effective_Y ==
8'd18) || (effective_X == 9'd30 && effective_Y == 8'd19) || (effective_X == 9'd30 &&
effective_Y == 8'd27) || (effective_X == 9'd30 && effective_Y == 8'd28) || (effective_X
== 9'd30 && effective_Y == 8'd37) || (effective_X == 9'd30 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd31 && effective_Y ==
8'd18) || (effective_X == 9'd31 && effective_Y == 8'd19) || (effective_X == 9'd31 &&
effective_Y == 8'd27) || (effective_X == 9'd31 && effective_Y == 8'd28) || (effective_X
== 9'd31 && effective_Y == 8'd37) || (effective_X == 9'd31 && effective_Y == 8'd38) ||

```







```

== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd37) || (effective_X == 9'd54 && effective_Y == 8'd38)

) begin

    pixel_colour = colourful ?
(((randNum_12b[4:2] ^ randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[4:2] ^
randNum_12b[4:2])) : 3'b111;

end

end

if (highscore == 12'd64) begin

    if((effective_X == 9'd29 && effective_Y ==
8'd18) || (effective_X == 9'd29 && effective_Y == 8'd19) || (effective_X == 9'd29 &&
effective_Y == 8'd20) || (effective_X == 9'd29 && effective_Y == 8'd21) || (effective_X
== 9'd29 && effective_Y == 8'd22) || (effective_X == 9'd29 && effective_Y == 8'd23) ||
(effective_X == 9'd29 && effective_Y == 8'd24) || (effective_X == 9'd29 && effective_Y
== 8'd25) || (effective_X == 9'd29 && effective_Y == 8'd26) || (effective_X == 9'd29 &&
effective_Y == 8'd27) || (effective_X == 9'd29 && effective_Y == 8'd28) || (effective_X
== 9'd29 && effective_Y == 8'd29) || (effective_X == 9'd29 && effective_Y == 8'd30) ||
(effective_X == 9'd29 && effective_Y == 8'd31) || (effective_X == 9'd29 && effective_Y
== 8'd32) || (effective_X == 9'd29 && effective_Y == 8'd33) || (effective_X == 9'd29 &&
effective_Y == 8'd34) || (effective_X == 9'd29 && effective_Y == 8'd35) || (effective_X
== 9'd29 && effective_Y == 8'd36) || (effective_X == 9'd29 && effective_Y == 8'd37) ||
(effective_X == 9'd29 && effective_Y == 8'd38) ||

        (effective_X == 9'd30 && effective_Y ==
8'd18) || (effective_X == 9'd30 && effective_Y == 8'd19) || (effective_X == 9'd30 &&
effective_Y == 8'd20) || (effective_X == 9'd30 && effective_Y == 8'd21) || (effective_X
== 9'd30 && effective_Y == 8'd22) || (effective_X == 9'd30 && effective_Y == 8'd23) ||
(effective_X == 9'd30 && effective_Y == 8'd24) || (effective_X == 9'd30 && effective_Y
== 8'd25) || (effective_X == 9'd30 && effective_Y == 8'd26) || (effective_X == 9'd30 &&
effective_Y == 8'd27) || (effective_X == 9'd30 && effective_Y == 8'd28) || (effective_X
== 9'd30 && effective_Y == 8'd29) || (effective_X == 9'd30 && effective_Y == 8'd30) ||
(effective_X == 9'd30 && effective_Y == 8'd31) || (effective_X == 9'd30 && effective_Y
== 8'd32) || (effective_X == 9'd30 && effective_Y == 8'd33) || (effective_X == 9'd30 &&
effective_Y == 8'd34) || (effective_X == 9'd30 && effective_Y == 8'd35) || (effective_X
== 9'd30 && effective_Y == 8'd36) || (effective_X == 9'd30 && effective_Y == 8'd37) ||
(effective_X == 9'd30 && effective_Y == 8'd38) ||

        (effective_X == 9'd31 && effective_Y ==
8'd18) || (effective_X == 9'd31 && effective_Y == 8'd19) || (effective_X == 9'd31 &&

```





```
(effective_X == 9'd52 && effective_Y ==  
8'd27) || (effective_X == 9'd52 && effective_Y == 8'd28) ||
```

```
(effective_X == 9'd53 && effective_Y ==  
8'd18) || (effective_X == 9'd53 && effective_Y == 8'd19) || (effective_X == 9'd53 &&  
effective_Y == 8'd20) || (effective_X == 9'd53 && effective_Y == 8'd21) || (effective_X  
== 9'd53 && effective_Y == 8'd22) || (effective_X == 9'd53 && effective_Y == 8'd23) ||  
(effective_X == 9'd53 && effective_Y == 8'd24) || (effective_X == 9'd53 && effective_Y  
== 8'd25) || (effective_X == 9'd53 && effective_Y == 8'd26) || (effective_X == 9'd53 &&  
effective_Y == 8'd27) || (effective_X == 9'd53 && effective_Y == 8'd28) || (effective_X  
== 9'd53 && effective_Y == 8'd29) || (effective_X == 9'd53 && effective_Y == 8'd30) ||  
(effective_X == 9'd53 && effective_Y == 8'd31) || (effective_X == 9'd53 && effective_Y  
== 8'd32) || (effective_X == 9'd53 && effective_Y == 8'd33) || (effective_X == 9'd53 &&  
effective_Y == 8'd34) || (effective_X == 9'd53 && effective_Y == 8'd35) || (effective_X  
== 9'd53 && effective_Y == 8'd36) || (effective_X == 9'd53 && effective_Y == 8'd37) ||  
(effective_X == 9'd53 && effective_Y == 8'd38) ||
```

```
(effective_X == 9'd54 && effective_Y ==  
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&  
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X  
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||  
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y  
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&  
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X  
== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||  
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y  
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&  
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X  
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||  
(effective_X == 9'd54 && effective_Y == 8'd38)
```

```
) begin
```

```
pixel_colour = colourful ?
```

```
((randNum_12b[7:5] ^ randNum_12b[8:6]) == 3'b0) ? 3'b111 : (randNum_12b[7:5] ^  
randNum_12b[8:6])) : 3'b111;
```

```
end
```

```
end
```

```
if (highscore == 12'd128) begin
```

```
if((effective_X == 9'd25 && effective_Y ==  
8'd18) || (effective_X == 9'd25 && effective_Y == 8'd19) || (effective_X == 9'd25 &&  
effective_Y == 8'd20) || (effective_X == 9'd25 && effective_Y == 8'd21) || (effective_X  
== 9'd25 && effective_Y == 8'd22) || (effective_X == 9'd25 && effective_Y == 8'd23) ||
```











```

== 9'd53 && effective_Y == 8'd36) || (effective_X == 9'd53 && effective_Y == 8'd37) ||
(effective_X == 9'd53 && effective_Y == 8'd38) ||

(effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

) begin

pixel_colour = colourful ?
(((randNum_12b[11:9] ^ randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^
randNum_12b[4:2])) : 3'b111;

end

end

if (highscore == 12'd256) begin

if((effective_X == 9'd15 && effective_Y ==
8'd18) || (effective_X == 9'd15 && effective_Y == 8'd19) || (effective_X == 9'd15 &&
effective_Y == 8'd27) || (effective_X == 9'd15 && effective_Y == 8'd28) || (effective_X
== 9'd15 && effective_Y == 8'd29) || (effective_X == 9'd15 && effective_Y == 8'd30) ||
(effective_X == 9'd15 && effective_Y == 8'd31) || (effective_X == 9'd15 && effective_Y
== 8'd32) || (effective_X == 9'd15 && effective_Y == 8'd33) || (effective_X == 9'd15 &&
effective_Y == 8'd34) || (effective_X == 9'd15 && effective_Y == 8'd35) || (effective_X
== 9'd15 && effective_Y == 8'd36) || (effective_X == 9'd15 && effective_Y == 8'd37) ||
(effective_X == 9'd15 && effective_Y == 8'd38) ||

(effective_X == 9'd16 && effective_Y ==
8'd18) || (effective_X == 9'd16 && effective_Y == 8'd19) || (effective_X == 9'd16 &&
effective_Y == 8'd27) || (effective_X == 9'd16 && effective_Y == 8'd28) || (effective_X
== 9'd16 && effective_Y == 8'd29) || (effective_X == 9'd16 && effective_Y == 8'd30) ||
(effective_X == 9'd16 && effective_Y == 8'd31) || (effective_X == 9'd16 && effective_Y
== 8'd32) || (effective_X == 9'd16 && effective_Y == 8'd33) || (effective_X == 9'd16 &&
effective_Y == 8'd34) || (effective_X == 9'd16 && effective_Y == 8'd35) || (effective_X

```













```

                                pixel_colour = colourful ?
(((randNum_12b[6:4] ^ randNum_12b[9:7]) == 3'b0) ? 3'b111 : (randNum_12b[6:4] ^
randNum_12b[9:7])) : 3'b111;

                                end

                                end

                                if (highscore == 12'd512) begin

                                        if((effective_X == 9'd15 && effective_Y ==
8'd18) || (effective_X == 9'd15 && effective_Y == 8'd19) || (effective_X == 9'd15 &&
effective_Y == 8'd20) || (effective_X == 9'd15 && effective_Y == 8'd21) || (effective_X
== 9'd15 && effective_Y == 8'd22) || (effective_X == 9'd15 && effective_Y == 8'd23) ||
(effective_X == 9'd15 && effective_Y == 8'd24) || (effective_X == 9'd15 && effective_Y
== 8'd25) || (effective_X == 9'd15 && effective_Y == 8'd26) || (effective_X == 9'd15 &&
effective_Y == 8'd27) || (effective_X == 9'd15 && effective_Y == 8'd28) || (effective_X
== 9'd15 && effective_Y == 8'd37) || (effective_X == 9'd15 && effective_Y == 8'd38) ||

                                                (effective_X == 9'd16 && effective_Y ==
8'd18) || (effective_X == 9'd16 && effective_Y == 8'd19) || (effective_X == 9'd16 &&
effective_Y == 8'd20) || (effective_X == 9'd16 && effective_Y == 8'd21) || (effective_X
== 9'd16 && effective_Y == 8'd22) || (effective_X == 9'd16 && effective_Y == 8'd23) ||
(effective_X == 9'd16 && effective_Y == 8'd24) || (effective_X == 9'd16 && effective_Y
== 8'd25) || (effective_X == 9'd16 && effective_Y == 8'd26) || (effective_X == 9'd16 &&
effective_Y == 8'd27) || (effective_X == 9'd16 && effective_Y == 8'd28) || (effective_X
== 9'd16 && effective_Y == 8'd37) || (effective_X == 9'd16 && effective_Y == 8'd38) ||

                                                (effective_X == 9'd17 && effective_Y ==
8'd18) || (effective_X == 9'd17 && effective_Y == 8'd19) || (effective_X == 9'd17 &&
effective_Y == 8'd27) || (effective_X == 9'd17 && effective_Y == 8'd28) || (effective_X
== 9'd17 && effective_Y == 8'd37) || (effective_X == 9'd17 && effective_Y == 8'd38) ||

                                                (effective_X == 9'd18 && effective_Y ==
8'd18) || (effective_X == 9'd18 && effective_Y == 8'd19) || (effective_X == 9'd18 &&
effective_Y == 8'd27) || (effective_X == 9'd18 && effective_Y == 8'd28) || (effective_X
== 9'd18 && effective_Y == 8'd37) || (effective_X == 9'd18 && effective_Y == 8'd38) ||

                                                (effective_X == 9'd19 && effective_Y ==
8'd18) || (effective_X == 9'd19 && effective_Y == 8'd19) || (effective_X == 9'd19 &&
effective_Y == 8'd27) || (effective_X == 9'd19 && effective_Y == 8'd28) || (effective_X
== 9'd19 && effective_Y == 8'd37) || (effective_X == 9'd19 && effective_Y == 8'd38) ||

                                                (effective_X == 9'd20 && effective_Y ==
8'd18) || (effective_X == 9'd20 && effective_Y == 8'd19) || (effective_X == 9'd20 &&
effective_Y == 8'd27) || (effective_X == 9'd20 && effective_Y == 8'd28) || (effective_X
== 9'd20 && effective_Y == 8'd37) || (effective_X == 9'd20 && effective_Y == 8'd38) ||

```







```

                (effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd37) || (effective_X == 9'd54 && effective_Y == 8'd38)

```

```

            ) begin

```

```

                pixel_colour = colourful ?

```

```

                (((randNum_12b[6:4] ^ randNum_12b[5:3]) == 3'b0) ? 3'b111 : (randNum_12b[6:4] ^
randNum_12b[5:3])) : 3'b111;

```

```

            end

```

```

        end

```

```

        if (highscore == 12'd1024) begin

```

```

                if((effective_X == 9'd11 && effective_Y ==
8'd18) || (effective_X == 9'd11 && effective_Y == 8'd19) || (effective_X == 9'd11 &&
effective_Y == 8'd20) || (effective_X == 9'd11 && effective_Y == 8'd21) || (effective_X
== 9'd11 && effective_Y == 8'd22) || (effective_X == 9'd11 && effective_Y == 8'd23) ||
(effective_X == 9'd11 && effective_Y == 8'd24) || (effective_X == 9'd11 && effective_Y
== 8'd25) || (effective_X == 9'd11 && effective_Y == 8'd26) || (effective_X == 9'd11 &&
effective_Y == 8'd27) || (effective_X == 9'd11 && effective_Y == 8'd28) || (effective_X
== 9'd11 && effective_Y == 8'd29) || (effective_X == 9'd11 && effective_Y == 8'd30) ||
(effective_X == 9'd11 && effective_Y == 8'd31) || (effective_X == 9'd11 && effective_Y
== 8'd32) || (effective_X == 9'd11 && effective_Y == 8'd33) || (effective_X == 9'd11 &&
effective_Y == 8'd34) || (effective_X == 9'd11 && effective_Y == 8'd35) || (effective_X
== 9'd11 && effective_Y == 8'd36) || (effective_X == 9'd11 && effective_Y == 8'd37) ||
(effective_X == 9'd11 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd12 && effective_Y ==
8'd18) || (effective_X == 9'd12 && effective_Y == 8'd19) || (effective_X == 9'd12 &&
effective_Y == 8'd20) || (effective_X == 9'd12 && effective_Y == 8'd21) || (effective_X
== 9'd12 && effective_Y == 8'd22) || (effective_X == 9'd12 && effective_Y == 8'd23) ||
(effective_X == 9'd12 && effective_Y == 8'd24) || (effective_X == 9'd12 && effective_Y
== 8'd25) || (effective_X == 9'd12 && effective_Y == 8'd26) || (effective_X == 9'd12 &&
effective_Y == 8'd27) || (effective_X == 9'd12 && effective_Y == 8'd28) || (effective_X
== 9'd12 && effective_Y == 8'd29) || (effective_X == 9'd12 && effective_Y == 8'd30) ||
(effective_X == 9'd12 && effective_Y == 8'd31) || (effective_X == 9'd12 && effective_Y
== 8'd32) || (effective_X == 9'd12 && effective_Y == 8'd33) || (effective_X == 9'd12 &&
effective_Y == 8'd34) || (effective_X == 9'd12 && effective_Y == 8'd35) || (effective_X

```













```

== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

) begin

    pixel_colour = colourful ?
(((randNum_12b[10:8] ^ randNum_12b[9:7]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^
randNum_12b[9:7])) : 3'b111;

end

end

if (highscore == 12'd2048) begin

    if((effective_X == 9'd1 && effective_Y ==
8'd18) || (effective_X == 9'd1 && effective_Y == 8'd19) || (effective_X == 9'd1 &&
effective_Y == 8'd27) || (effective_X == 9'd1 && effective_Y == 8'd28) || (effective_X ==
9'd1 && effective_Y == 8'd29) || (effective_X == 9'd1 && effective_Y == 8'd30) ||
(effective_X == 9'd1 && effective_Y == 8'd31) || (effective_X == 9'd1 && effective_Y ==
8'd32) || (effective_X == 9'd1 && effective_Y == 8'd33) || (effective_X == 9'd1 &&
effective_Y == 8'd34) || (effective_X == 9'd1 && effective_Y == 8'd35) || (effective_X ==
9'd1 && effective_Y == 8'd36) || (effective_X == 9'd1 && effective_Y == 8'd37) ||
(effective_X == 9'd1 && effective_Y == 8'd38) ||

        (effective_X == 9'd2 && effective_Y ==
8'd18) || (effective_X == 9'd2 && effective_Y == 8'd19) || (effective_X == 9'd2 &&
effective_Y == 8'd27) || (effective_X == 9'd2 && effective_Y == 8'd28) || (effective_X ==
9'd2 && effective_Y == 8'd29) || (effective_X == 9'd2 && effective_Y == 8'd30) ||
(effective_X == 9'd2 && effective_Y == 8'd31) || (effective_X == 9'd2 && effective_Y ==
8'd32) || (effective_X == 9'd2 && effective_Y == 8'd33) || (effective_X == 9'd2 &&
effective_Y == 8'd34) || (effective_X == 9'd2 && effective_Y == 8'd35) || (effective_X ==
9'd2 && effective_Y == 8'd36) || (effective_X == 9'd2 && effective_Y == 8'd37) ||
(effective_X == 9'd2 && effective_Y == 8'd38) ||

        (effective_X == 9'd3 && effective_Y ==
8'd18) || (effective_X == 9'd3 && effective_Y == 8'd19) || (effective_X == 9'd3 &&
effective_Y == 8'd27) || (effective_X == 9'd3 && effective_Y == 8'd28) || (effective_X ==
9'd3 && effective_Y == 8'd37) || (effective_X == 9'd3 && effective_Y == 8'd38) ||

        (effective_X == 9'd4 && effective_Y ==
8'd18) || (effective_X == 9'd4 && effective_Y == 8'd19) || (effective_X == 9'd4 &&
effective_Y == 8'd27) || (effective_X == 9'd4 && effective_Y == 8'd28) || (effective_X ==
9'd4 && effective_Y == 8'd37) || (effective_X == 9'd4 && effective_Y == 8'd38) ||

```















```

(effective_X == 9'd53 && effective_Y == 8'd24) || (effective_X == 9'd53 && effective_Y
== 8'd25) || (effective_X == 9'd53 && effective_Y == 8'd26) || (effective_X == 9'd53 &&
effective_Y == 8'd27) || (effective_X == 9'd53 && effective_Y == 8'd28) || (effective_X
== 9'd53 && effective_Y == 8'd29) || (effective_X == 9'd53 && effective_Y == 8'd30) ||
(effective_X == 9'd53 && effective_Y == 8'd31) || (effective_X == 9'd53 && effective_Y
== 8'd32) || (effective_X == 9'd53 && effective_Y == 8'd33) || (effective_X == 9'd53 &&
effective_Y == 8'd34) || (effective_X == 9'd53 && effective_Y == 8'd35) || (effective_X
== 9'd53 && effective_Y == 8'd36) || (effective_X == 9'd53 && effective_Y == 8'd37) ||
(effective_X == 9'd53 && effective_Y == 8'd38) ||

(effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

) begin

pixel_colour = colourful ?
(((randNum_12b[10:8] ^ randNum_12b[6:4]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^
randNum_12b[6:4])) : 3'b111;

end

end

// if (highscore == 12'd4096) begin
// if(effective_X == 6'd1 && effective_Y == 6'd1 ||
// effective_X == 6'd2 && effective_Y ==
6'd2 ||
// effective_X == 6'd3 && effective_Y ==
6'd3 ||
// effective_X == 6'd4 && effective_Y ==
6'd4 ||

```

```

//                effective_X == 6'd5 && effective_Y ==
6'd5 ||

//                effective_X == 6'd6 && effective_Y ==
6'd6 ||

//                effective_X == 6'd7 && effective_Y ==
6'd7 ||

//                effective_X == 6'd8 && effective_Y ==
6'd8 ||

//                effective_X == 6'd9 && effective_Y ==
6'd9 ||

//                effective_X == 6'd10 && effective_Y ==
6'd10 ||

//                effective_X == 6'd11 && effective_Y ==
6'd11 ||

//                effective_X == 6'd12 && effective_Y ==
6'd12) begin
//                pixel_colour = 3'b111;
//                end
//                end
//                end
end

if (screen_X <= 9'd239 && screen_Y <= 9'd239) begin
    if (screen_X == 9'd0 || // For drawing borders and lines
        screen_X == 9'd1 ||
        screen_X == 9'd2 ||
        screen_X == 9'd60 ||
        screen_X == 9'd61 ||
        screen_X == 9'd119 ||
        screen_X == 9'd120 ||
        screen_X == 9'd178 ||
        screen_X == 9'd179 ||

```

```

        screen_X == 9'd237 ||
        screen_X == 9'd238 ||
        screen_X == 9'd239 ||
        screen_Y == 8'd0  ||
        screen_Y == 8'd1  ||
        screen_Y == 8'd2  ||
        screen_Y == 8'd60 ||
        screen_Y == 8'd61 ||
        screen_Y == 8'd119 ||
        screen_Y == 8'd120 ||
        screen_Y == 8'd178 ||
        screen_Y == 8'd179 ||
        screen_Y == 8'd237 ||
        screen_Y == 8'd238 ||
        screen_Y == 8'd239) begin
            pixel_colour = colourful ? (((randNum_12b[9:7] ^
randNum_12b[11:9]) == 3'b0) ? 3'b111 : (randNum_12b[9:7] ^ randNum_12b[11:9])) :
3'b111;

        end
    else begin
        if (screen_X <= 2'd3) begin
            effective_X = 6'd0;
        end
        else begin
            effective_X = ((screen_X - 2'd3) -
((gameBoard_cur_X) * (6'd59)));
        end
        if (screen_Y <= 2'd3) begin
            effective_Y = 6'd0;

```

```

end
else begin
    effective_Y = ((screen_Y - 2'd3) -
((gameBoard_cur_Y) * (6'd59)));
end
if (gameBoard_cur_Value == 12'd0) begin
end
if (gameBoard_cur_Value == 12'd2) begin
    if((effective_X == 9'd43 && effective_Y ==
8'd18) || (effective_X == 9'd43 && effective_Y == 8'd19) || (effective_X == 9'd43 &&
effective_Y == 8'd27) || (effective_X == 9'd43 && effective_Y == 8'd28) || (effective_X
== 9'd43 && effective_Y == 8'd29) || (effective_X == 9'd43 && effective_Y == 8'd30) ||
(effective_X == 9'd43 && effective_Y == 8'd31) || (effective_X == 9'd43 && effective_Y
== 8'd32) || (effective_X == 9'd43 && effective_Y == 8'd33) || (effective_X == 9'd43 &&
effective_Y == 8'd34) || (effective_X == 9'd43 && effective_Y == 8'd35) || (effective_X
== 9'd43 && effective_Y == 8'd36) || (effective_X == 9'd43 && effective_Y == 8'd37) ||
(effective_X == 9'd43 && effective_Y == 8'd38) ||

(effective_X == 9'd44 && effective_Y ==
8'd18) || (effective_X == 9'd44 && effective_Y == 8'd19) || (effective_X == 9'd44 &&
effective_Y == 8'd27) || (effective_X == 9'd44 && effective_Y == 8'd28) || (effective_X
== 9'd44 && effective_Y == 8'd29) || (effective_X == 9'd44 && effective_Y == 8'd30) ||
(effective_X == 9'd44 && effective_Y == 8'd31) || (effective_X == 9'd44 && effective_Y
== 8'd32) || (effective_X == 9'd44 && effective_Y == 8'd33) || (effective_X == 9'd44 &&
effective_Y == 8'd34) || (effective_X == 9'd44 && effective_Y == 8'd35) || (effective_X
== 9'd44 && effective_Y == 8'd36) || (effective_X == 9'd44 && effective_Y == 8'd37) ||
(effective_X == 9'd44 && effective_Y == 8'd38) ||

(effective_X == 9'd45 && effective_Y ==
8'd18) || (effective_X == 9'd45 && effective_Y == 8'd19) || (effective_X == 9'd45 &&
effective_Y == 8'd27) || (effective_X == 9'd45 && effective_Y == 8'd28) || (effective_X
== 9'd45 && effective_Y == 8'd37) || (effective_X == 9'd45 && effective_Y == 8'd38) ||

(effective_X == 9'd46 && effective_Y ==
8'd18) || (effective_X == 9'd46 && effective_Y == 8'd19) || (effective_X == 9'd46 &&
effective_Y == 8'd27) || (effective_X == 9'd46 && effective_Y == 8'd28) || (effective_X
== 9'd46 && effective_Y == 8'd37) || (effective_X == 9'd46 && effective_Y == 8'd38) ||

(effective_X == 9'd47 && effective_Y ==
8'd18) || (effective_X == 9'd47 && effective_Y == 8'd19) || (effective_X == 9'd47 &&

```





```

                                pixel_colour = colourful ?
(((randNum_12b[7:5] ^ randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[7:5] ^
randNum_12b[4:2])) : 3'b111;

                                end

                                end

                                if (gameBoard_cur_Value == 12'd4) begin

                                        if((effective_X == 9'd43 && effective_Y ==
8'd18) || (effective_X == 9'd43 && effective_Y == 8'd19) || (effective_X == 9'd43 &&
effective_Y == 8'd20) || (effective_X == 9'd43 && effective_Y == 8'd21) || (effective_X
== 9'd43 && effective_Y == 8'd22) || (effective_X == 9'd43 && effective_Y == 8'd23) ||
(effective_X == 9'd43 && effective_Y == 8'd24) || (effective_X == 9'd43 && effective_Y
== 8'd25) || (effective_X == 9'd43 && effective_Y == 8'd26) || (effective_X == 9'd43 &&
effective_Y == 8'd27) || (effective_X == 9'd43 && effective_Y == 8'd28) ||

                                                (effective_X == 9'd44 && effective_Y ==
8'd18) || (effective_X == 9'd44 && effective_Y == 8'd19) || (effective_X == 9'd44 &&
effective_Y == 8'd20) || (effective_X == 9'd44 && effective_Y == 8'd21) || (effective_X
== 9'd44 && effective_Y == 8'd22) || (effective_X == 9'd44 && effective_Y == 8'd23) ||
(effective_X == 9'd44 && effective_Y == 8'd24) || (effective_X == 9'd44 && effective_Y
== 8'd25) || (effective_X == 9'd44 && effective_Y == 8'd26) || (effective_X == 9'd44 &&
effective_Y == 8'd27) || (effective_X == 9'd44 && effective_Y == 8'd28) ||

                                                (effective_X == 9'd45 && effective_Y ==
8'd27) || (effective_X == 9'd45 && effective_Y == 8'd28) ||

                                                (effective_X == 9'd46 && effective_Y ==
8'd27) || (effective_X == 9'd46 && effective_Y == 8'd28) ||

                                                (effective_X == 9'd47 && effective_Y ==
8'd27) || (effective_X == 9'd47 && effective_Y == 8'd28) ||

                                                (effective_X == 9'd48 && effective_Y ==
8'd27) || (effective_X == 9'd48 && effective_Y == 8'd28) ||

                                                (effective_X == 9'd49 && effective_Y ==
8'd27) || (effective_X == 9'd49 && effective_Y == 8'd28) ||

                                                (effective_X == 9'd50 && effective_Y ==
8'd27) || (effective_X == 9'd50 && effective_Y == 8'd28) ||

                                                (effective_X == 9'd51 && effective_Y ==
8'd27) || (effective_X == 9'd51 && effective_Y == 8'd28) ||

                                                (effective_X == 9'd52 && effective_Y ==
8'd27) || (effective_X == 9'd52 && effective_Y == 8'd28) ||

```

```

                (effective_X == 9'd53 && effective_Y ==
8'd18) || (effective_X == 9'd53 && effective_Y == 8'd19) || (effective_X == 9'd53 &&
effective_Y == 8'd20) || (effective_X == 9'd53 && effective_Y == 8'd21) || (effective_X
== 9'd53 && effective_Y == 8'd22) || (effective_X == 9'd53 && effective_Y == 8'd23) ||
(effective_X == 9'd53 && effective_Y == 8'd24) || (effective_X == 9'd53 && effective_Y
== 8'd25) || (effective_X == 9'd53 && effective_Y == 8'd26) || (effective_X == 9'd53 &&
effective_Y == 8'd27) || (effective_X == 9'd53 && effective_Y == 8'd28) || (effective_X
== 9'd53 && effective_Y == 8'd29) || (effective_X == 9'd53 && effective_Y == 8'd30) ||
(effective_X == 9'd53 && effective_Y == 8'd31) || (effective_X == 9'd53 && effective_Y
== 8'd32) || (effective_X == 9'd53 && effective_Y == 8'd33) || (effective_X == 9'd53 &&
effective_Y == 8'd34) || (effective_X == 9'd53 && effective_Y == 8'd35) || (effective_X
== 9'd53 && effective_Y == 8'd36) || (effective_X == 9'd53 && effective_Y == 8'd37) ||
(effective_X == 9'd53 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

```

```

            ) begin

```

```

                pixel_colour = colourful ?

```

```

                (((randNum_12b[10:8] ^ randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^
randNum_12b[4:2])) : 3'b111;

```

```

            end

```

```

        end

```

```

        if (gameBoard_cur_Value == 12'd8) begin

```

```

                if((effective_X == 9'd43 && effective_Y ==
8'd18) || (effective_X == 9'd43 && effective_Y == 8'd19) || (effective_X == 9'd43 &&
effective_Y == 8'd20) || (effective_X == 9'd43 && effective_Y == 8'd21) || (effective_X
== 9'd43 && effective_Y == 8'd22) || (effective_X == 9'd43 && effective_Y == 8'd23) ||
(effective_X == 9'd43 && effective_Y == 8'd24) || (effective_X == 9'd43 && effective_Y
== 8'd25) || (effective_X == 9'd43 && effective_Y == 8'd26) || (effective_X == 9'd43 &&

```









```

                (effective_X == 9'd50 && effective_Y ==
8'd18) || (effective_X == 9'd50 && effective_Y == 8'd19) || (effective_X == 9'd50 &&
effective_Y == 8'd27) || (effective_X == 9'd50 && effective_Y == 8'd28) || (effective_X
== 9'd50 && effective_Y == 8'd37) || (effective_X == 9'd50 && effective_Y == 8'd38)||

```

```

                (effective_X == 9'd51 && effective_Y ==
8'd18) || (effective_X == 9'd51 && effective_Y == 8'd19) || (effective_X == 9'd51 &&
effective_Y == 8'd27) || (effective_X == 9'd51 && effective_Y == 8'd28) || (effective_X
== 9'd51 && effective_Y == 8'd37) || (effective_X == 9'd51 && effective_Y == 8'd38)||

```

```

                (effective_X == 9'd52 && effective_Y ==
8'd18) || (effective_X == 9'd52 && effective_Y == 8'd19) || (effective_X == 9'd52 &&
effective_Y == 8'd27) || (effective_X == 9'd52 && effective_Y == 8'd28) || (effective_X
== 9'd52 && effective_Y == 8'd37) || (effective_X == 9'd52 && effective_Y == 8'd38)||

```

```

                (effective_X == 9'd53 && effective_Y ==
8'd18) || (effective_X == 9'd53 && effective_Y == 8'd19) || (effective_X == 9'd53 &&
effective_Y == 8'd27) || (effective_X == 9'd53 && effective_Y == 8'd28) || (effective_X
== 9'd53 && effective_Y == 8'd29) || (effective_X == 9'd53 && effective_Y == 8'd30) ||
(effective_X == 9'd53 && effective_Y == 8'd31) || (effective_X == 9'd53 && effective_Y
== 8'd32) || (effective_X == 9'd53 && effective_Y == 8'd33) || (effective_X == 9'd53 &&
effective_Y == 8'd34) || (effective_X == 9'd53 && effective_Y == 8'd35) || (effective_X
== 9'd53 && effective_Y == 8'd36) || (effective_X == 9'd53 && effective_Y == 8'd37) ||
(effective_X == 9'd53 && effective_Y == 8'd38)||

```

```

                (effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

```

```

            ) begin

```

```

                pixel_colour = colourful ?

```

```

                (((randNum_12b[6:4] ^ randNum_12b[7:5]) == 3'b0) ? 3'b111 : (randNum_12b[6:4] ^
randNum_12b[7:5])) : 3'b111;

```

```

            end

```

```

        end

```

```

        if (gameBoard_cur_Value == 12'd32) begin

```

[illegible]







```

                (effective_X == 9'd53 && effective_Y ==
8'd18) || (effective_X == 9'd53 && effective_Y == 8'd19) || (effective_X == 9'd53 &&
effective_Y == 8'd20) || (effective_X == 9'd53 && effective_Y == 8'd21) || (effective_X
== 9'd53 && effective_Y == 8'd22) || (effective_X == 9'd53 && effective_Y == 8'd23) ||
(effective_X == 9'd53 && effective_Y == 8'd24) || (effective_X == 9'd53 && effective_Y
== 8'd25) || (effective_X == 9'd53 && effective_Y == 8'd26) || (effective_X == 9'd53 &&
effective_Y == 8'd27) || (effective_X == 9'd53 && effective_Y == 8'd28) || (effective_X
== 9'd53 && effective_Y == 8'd37) || (effective_X == 9'd53 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd37) || (effective_X == 9'd54 && effective_Y == 8'd38)

```

```

            ) begin

```

```

                pixel_colour = colourful ?

```

```

                (((randNum_12b[4:2] ^ randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[4:2] ^
randNum_12b[4:2])) : 3'b111;

```

```

            end

```

```

        end

```

```

        if (gameBoard_cur_Value == 12'd64) begin

```

```

                if((effective_X == 9'd29 && effective_Y ==
8'd18) || (effective_X == 9'd29 && effective_Y == 8'd19) || (effective_X == 9'd29 &&
effective_Y == 8'd20) || (effective_X == 9'd29 && effective_Y == 8'd21) || (effective_X
== 9'd29 && effective_Y == 8'd22) || (effective_X == 9'd29 && effective_Y == 8'd23) ||
(effective_X == 9'd29 && effective_Y == 8'd24) || (effective_X == 9'd29 && effective_Y
== 8'd25) || (effective_X == 9'd29 && effective_Y == 8'd26) || (effective_X == 9'd29 &&
effective_Y == 8'd27) || (effective_X == 9'd29 && effective_Y == 8'd28) || (effective_X
== 9'd29 && effective_Y == 8'd29) || (effective_X == 9'd29 && effective_Y == 8'd30) ||
(effective_X == 9'd29 && effective_Y == 8'd31) || (effective_X == 9'd29 && effective_Y
== 8'd32) || (effective_X == 9'd29 && effective_Y == 8'd33) || (effective_X == 9'd29 &&
effective_Y == 8'd34) || (effective_X == 9'd29 && effective_Y == 8'd35) || (effective_X
== 9'd29 && effective_Y == 8'd36) || (effective_X == 9'd29 && effective_Y == 8'd37) ||
(effective_X == 9'd29 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd30 && effective_Y ==
8'd18) || (effective_X == 9'd30 && effective_Y == 8'd19) || (effective_X == 9'd30 &&
effective_Y == 8'd20) || (effective_X == 9'd30 && effective_Y == 8'd21) || (effective_X

```







```

                                pixel_colour = colourful ?
(((randNum_12b[7:5] ^ randNum_12b[8:6]) == 3'b0) ? 3'b111 : (randNum_12b[7:5] ^
randNum_12b[8:6])) : 3'b111;

```

```

                                end

```

```

                                end

```

```

                                if (gameBoard_cur_Value == 12'd128) begin

```

```

                                    if((effective_X == 9'd25 && effective_Y ==
8'd18) || (effective_X == 9'd25 && effective_Y == 8'd19) || (effective_X == 9'd25 &&
effective_Y == 8'd20) || (effective_X == 9'd25 && effective_Y == 8'd21) || (effective_X
== 9'd25 && effective_Y == 8'd22) || (effective_X == 9'd25 && effective_Y == 8'd23) ||
(effective_X == 9'd25 && effective_Y == 8'd24) || (effective_X == 9'd25 && effective_Y
== 8'd25) || (effective_X == 9'd25 && effective_Y == 8'd26) || (effective_X == 9'd25 &&
effective_Y == 8'd27) || (effective_X == 9'd25 && effective_Y == 8'd28) || (effective_X
== 9'd25 && effective_Y == 8'd29) || (effective_X == 9'd25 && effective_Y == 8'd30) ||
(effective_X == 9'd25 && effective_Y == 8'd31) || (effective_X == 9'd25 && effective_Y
== 8'd32) || (effective_X == 9'd25 && effective_Y == 8'd33) || (effective_X == 9'd25 &&
effective_Y == 8'd34) || (effective_X == 9'd25 && effective_Y == 8'd35) || (effective_X
== 9'd25 && effective_Y == 8'd36) || (effective_X == 9'd25 && effective_Y == 8'd37) ||
(effective_X == 9'd25 && effective_Y == 8'd38) ||

```

```

                                    (effective_X == 9'd26 && effective_Y ==
8'd18) || (effective_X == 9'd26 && effective_Y == 8'd19) || (effective_X == 9'd26 &&
effective_Y == 8'd20) || (effective_X == 9'd26 && effective_Y == 8'd21) || (effective_X
== 9'd26 && effective_Y == 8'd22) || (effective_X == 9'd26 && effective_Y == 8'd23) ||
(effective_X == 9'd26 && effective_Y == 8'd24) || (effective_X == 9'd26 && effective_Y
== 8'd25) || (effective_X == 9'd26 && effective_Y == 8'd26) || (effective_X == 9'd26 &&
effective_Y == 8'd27) || (effective_X == 9'd26 && effective_Y == 8'd28) || (effective_X
== 9'd26 && effective_Y == 8'd29) || (effective_X == 9'd26 && effective_Y == 8'd30) ||
(effective_X == 9'd26 && effective_Y == 8'd31) || (effective_X == 9'd26 && effective_Y
== 8'd32) || (effective_X == 9'd26 && effective_Y == 8'd33) || (effective_X == 9'd26 &&
effective_Y == 8'd34) || (effective_X == 9'd26 && effective_Y == 8'd35) || (effective_X
== 9'd26 && effective_Y == 8'd36) || (effective_X == 9'd26 && effective_Y == 8'd37) ||
(effective_X == 9'd26 && effective_Y == 8'd38) ||

```

```

                                    (effective_X == 9'd29 && effective_Y ==
8'd18) || (effective_X == 9'd29 && effective_Y == 8'd19) || (effective_X == 9'd29 &&
effective_Y == 8'd27) || (effective_X == 9'd29 && effective_Y == 8'd28) || (effective_X
== 9'd29 && effective_Y == 8'd29) || (effective_X == 9'd29 && effective_Y == 8'd30) ||
(effective_X == 9'd29 && effective_Y == 8'd31) || (effective_X == 9'd29 && effective_Y
== 8'd32) || (effective_X == 9'd29 && effective_Y == 8'd33) || (effective_X == 9'd29 &&
effective_Y == 8'd34) || (effective_X == 9'd29 && effective_Y == 8'd35) || (effective_X

```









```

                (effective_X == 9'd53 && effective_Y ==
8'd18) || (effective_X == 9'd53 && effective_Y == 8'd19) || (effective_X == 9'd53 &&
effective_Y == 8'd20) || (effective_X == 9'd53 && effective_Y == 8'd21) || (effective_X
== 9'd53 && effective_Y == 8'd22) || (effective_X == 9'd53 && effective_Y == 8'd23) ||
(effective_X == 9'd53 && effective_Y == 8'd24) || (effective_X == 9'd53 && effective_Y
== 8'd25) || (effective_X == 9'd53 && effective_Y == 8'd26) || (effective_X == 9'd53 &&
effective_Y == 8'd27) || (effective_X == 9'd53 && effective_Y == 8'd28) || (effective_X
== 9'd53 && effective_Y == 8'd29) || (effective_X == 9'd53 && effective_Y == 8'd30) ||
(effective_X == 9'd53 && effective_Y == 8'd31) || (effective_X == 9'd53 && effective_Y
== 8'd32) || (effective_X == 9'd53 && effective_Y == 8'd33) || (effective_X == 9'd53 &&
effective_Y == 8'd34) || (effective_X == 9'd53 && effective_Y == 8'd35) || (effective_X
== 9'd53 && effective_Y == 8'd36) || (effective_X == 9'd53 && effective_Y == 8'd37) ||
(effective_X == 9'd53 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

```

```

            ) begin

```

```

                pixel_colour = colourful ?

```

```

                (((randNum_12b[11:9] ^ randNum_12b[4:2]) == 3'b0) ? 3'b111 : (randNum_12b[11:9] ^
randNum_12b[4:2])) : 3'b111;

```

```

            end

```

```

        end

```

```

        if (gameBoard_cur_Value == 12'd256) begin

```

```

                if((effective_X == 9'd15 && effective_Y ==
8'd18) || (effective_X == 9'd15 && effective_Y == 8'd19) || (effective_X == 9'd15 &&
effective_Y == 8'd27) || (effective_X == 9'd15 && effective_Y == 8'd28) || (effective_X
== 9'd15 && effective_Y == 8'd29) || (effective_X == 9'd15 && effective_Y == 8'd30) ||
(effective_X == 9'd15 && effective_Y == 8'd31) || (effective_X == 9'd15 && effective_Y
== 8'd32) || (effective_X == 9'd15 && effective_Y == 8'd33) || (effective_X == 9'd15 &&

```













```

(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

    ) begin

        pixel_colour = colourful ?
(((randNum_12b[6:4] ^ randNum_12b[9:7]) == 3'b0) ? 3'b111 : (randNum_12b[6:4] ^
randNum_12b[9:7])) : 3'b111;

    end

end

if (gameBoard_cur_Value == 12'd512) begin

    if((effective_X == 9'd15 && effective_Y ==
8'd18) || (effective_X == 9'd15 && effective_Y == 8'd19) || (effective_X == 9'd15 &&
effective_Y == 8'd20) || (effective_X == 9'd15 && effective_Y == 8'd21) || (effective_X
== 9'd15 && effective_Y == 8'd22) || (effective_X == 9'd15 && effective_Y == 8'd23) ||
(effective_X == 9'd15 && effective_Y == 8'd24) || (effective_X == 9'd15 && effective_Y
== 8'd25) || (effective_X == 9'd15 && effective_Y == 8'd26) || (effective_X == 9'd15 &&
effective_Y == 8'd27) || (effective_X == 9'd15 && effective_Y == 8'd28) || (effective_X
== 9'd15 && effective_Y == 8'd37) || (effective_X == 9'd15 && effective_Y == 8'd38) ||

        (effective_X == 9'd16 && effective_Y ==
8'd18) || (effective_X == 9'd16 && effective_Y == 8'd19) || (effective_X == 9'd16 &&
effective_Y == 8'd20) || (effective_X == 9'd16 && effective_Y == 8'd21) || (effective_X
== 9'd16 && effective_Y == 8'd22) || (effective_X == 9'd16 && effective_Y == 8'd23) ||
(effective_X == 9'd16 && effective_Y == 8'd24) || (effective_X == 9'd16 && effective_Y
== 8'd25) || (effective_X == 9'd16 && effective_Y == 8'd26) || (effective_X == 9'd16 &&
effective_Y == 8'd27) || (effective_X == 9'd16 && effective_Y == 8'd28) || (effective_X
== 9'd16 && effective_Y == 8'd37) || (effective_X == 9'd16 && effective_Y == 8'd38) ||

        (effective_X == 9'd17 && effective_Y ==
8'd18) || (effective_X == 9'd17 && effective_Y == 8'd19) || (effective_X == 9'd17 &&
effective_Y == 8'd27) || (effective_X == 9'd17 && effective_Y == 8'd28) || (effective_X
== 9'd17 && effective_Y == 8'd37) || (effective_X == 9'd17 && effective_Y == 8'd38) ||

        (effective_X == 9'd18 && effective_Y ==
8'd18) || (effective_X == 9'd18 && effective_Y == 8'd19) || (effective_X == 9'd18 &&
effective_Y == 8'd27) || (effective_X == 9'd18 && effective_Y == 8'd28) || (effective_X
== 9'd18 && effective_Y == 8'd37) || (effective_X == 9'd18 && effective_Y == 8'd38) ||

        (effective_X == 9'd19 && effective_Y ==
8'd18) || (effective_X == 9'd19 && effective_Y == 8'd19) || (effective_X == 9'd19 &&

```







```

                (effective_X == 9'd53 && effective_Y ==
8'd18) || (effective_X == 9'd53 && effective_Y == 8'd19) || (effective_X == 9'd53 &&
effective_Y == 8'd20) || (effective_X == 9'd53 && effective_Y == 8'd21) || (effective_X
== 9'd53 && effective_Y == 8'd22) || (effective_X == 9'd53 && effective_Y == 8'd23) ||
(effective_X == 9'd53 && effective_Y == 8'd24) || (effective_X == 9'd53 && effective_Y
== 8'd25) || (effective_X == 9'd53 && effective_Y == 8'd26) || (effective_X == 9'd53 &&
effective_Y == 8'd27) || (effective_X == 9'd53 && effective_Y == 8'd28) || (effective_X
== 9'd53 && effective_Y == 8'd37) || (effective_X == 9'd53 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd37) || (effective_X == 9'd54 && effective_Y == 8'd38)

```

```

            ) begin

```

```

                pixel_colour = colourful ?

```

```

                (((randNum_12b[6:4] ^ randNum_12b[5:3]) == 3'b0) ? 3'b111 : (randNum_12b[6:4] ^
randNum_12b[5:3])) : 3'b111;

```

```

            end

```

```

        end

```

```

        if (gameBoard_cur_Value == 12'd1024) begin

```

```

                if((effective_X == 9'd11 && effective_Y ==
8'd18) || (effective_X == 9'd11 && effective_Y == 8'd19) || (effective_X == 9'd11 &&
effective_Y == 8'd20) || (effective_X == 9'd11 && effective_Y == 8'd21) || (effective_X
== 9'd11 && effective_Y == 8'd22) || (effective_X == 9'd11 && effective_Y == 8'd23) ||
(effective_X == 9'd11 && effective_Y == 8'd24) || (effective_X == 9'd11 && effective_Y
== 8'd25) || (effective_X == 9'd11 && effective_Y == 8'd26) || (effective_X == 9'd11 &&
effective_Y == 8'd27) || (effective_X == 9'd11 && effective_Y == 8'd28) || (effective_X
== 9'd11 && effective_Y == 8'd29) || (effective_X == 9'd11 && effective_Y == 8'd30) ||
(effective_X == 9'd11 && effective_Y == 8'd31) || (effective_X == 9'd11 && effective_Y
== 8'd32) || (effective_X == 9'd11 && effective_Y == 8'd33) || (effective_X == 9'd11 &&
effective_Y == 8'd34) || (effective_X == 9'd11 && effective_Y == 8'd35) || (effective_X
== 9'd11 && effective_Y == 8'd36) || (effective_X == 9'd11 && effective_Y == 8'd37) ||
(effective_X == 9'd11 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd12 && effective_Y ==
8'd18) || (effective_X == 9'd12 && effective_Y == 8'd19) || (effective_X == 9'd12 &&
effective_Y == 8'd20) || (effective_X == 9'd12 && effective_Y == 8'd21) || (effective_X

```













```

(effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

```

```

) begin

```

```

    pixel_colour = colourful ?

```

```

(((randNum_12b[10:8] ^ randNum_12b[9:7]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^
randNum_12b[9:7])) : 3'b111;

```

```

end

```

```

end

```

```

if (gameBoard_cur_Value == 12'd2048) begin

```

```

    if((effective_X == 9'd1 && effective_Y ==
8'd18) || (effective_X == 9'd1 && effective_Y == 8'd19) || (effective_X == 9'd1 &&
effective_Y == 8'd27) || (effective_X == 9'd1 && effective_Y == 8'd28) || (effective_X ==
9'd1 && effective_Y == 8'd29) || (effective_X == 9'd1 && effective_Y == 8'd30) ||
(effective_X == 9'd1 && effective_Y == 8'd31) || (effective_X == 9'd1 && effective_Y ==
8'd32) || (effective_X == 9'd1 && effective_Y == 8'd33) || (effective_X == 9'd1 &&
effective_Y == 8'd34) || (effective_X == 9'd1 && effective_Y == 8'd35) || (effective_X ==
9'd1 && effective_Y == 8'd36) || (effective_X == 9'd1 && effective_Y == 8'd37) ||
(effective_X == 9'd1 && effective_Y == 8'd38) ||

```

```

(effective_X == 9'd2 && effective_Y ==
8'd18) || (effective_X == 9'd2 && effective_Y == 8'd19) || (effective_X == 9'd2 &&
effective_Y == 8'd27) || (effective_X == 9'd2 && effective_Y == 8'd28) || (effective_X ==
9'd2 && effective_Y == 8'd29) || (effective_X == 9'd2 && effective_Y == 8'd30) ||
(effective_X == 9'd2 && effective_Y == 8'd31) || (effective_X == 9'd2 && effective_Y ==
8'd32) || (effective_X == 9'd2 && effective_Y == 8'd33) || (effective_X == 9'd2 &&
effective_Y == 8'd34) || (effective_X == 9'd2 && effective_Y == 8'd35) || (effective_X ==
9'd2 && effective_Y == 8'd36) || (effective_X == 9'd2 && effective_Y == 8'd37) ||
(effective_X == 9'd2 && effective_Y == 8'd38) ||

```















```

                (effective_X == 9'd52 && effective_Y ==
8'd18) || (effective_X == 9'd52 && effective_Y == 8'd19) || (effective_X == 9'd52 &&
effective_Y == 8'd27) || (effective_X == 9'd52 && effective_Y == 8'd28) || (effective_X
== 9'd52 && effective_Y == 8'd37) || (effective_X == 9'd52 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd53 && effective_Y ==
8'd18) || (effective_X == 9'd53 && effective_Y == 8'd19) || (effective_X == 9'd53 &&
effective_Y == 8'd20) || (effective_X == 9'd53 && effective_Y == 8'd21) || (effective_X
== 9'd53 && effective_Y == 8'd22) || (effective_X == 9'd53 && effective_Y == 8'd23) ||
(effective_X == 9'd53 && effective_Y == 8'd24) || (effective_X == 9'd53 && effective_Y
== 8'd25) || (effective_X == 9'd53 && effective_Y == 8'd26) || (effective_X == 9'd53 &&
effective_Y == 8'd27) || (effective_X == 9'd53 && effective_Y == 8'd28) || (effective_X
== 9'd53 && effective_Y == 8'd29) || (effective_X == 9'd53 && effective_Y == 8'd30) ||
(effective_X == 9'd53 && effective_Y == 8'd31) || (effective_X == 9'd53 && effective_Y
== 8'd32) || (effective_X == 9'd53 && effective_Y == 8'd33) || (effective_X == 9'd53 &&
effective_Y == 8'd34) || (effective_X == 9'd53 && effective_Y == 8'd35) || (effective_X
== 9'd53 && effective_Y == 8'd36) || (effective_X == 9'd53 && effective_Y == 8'd37) ||
(effective_X == 9'd53 && effective_Y == 8'd38) ||

```

```

                (effective_X == 9'd54 && effective_Y ==
8'd18) || (effective_X == 9'd54 && effective_Y == 8'd19) || (effective_X == 9'd54 &&
effective_Y == 8'd20) || (effective_X == 9'd54 && effective_Y == 8'd21) || (effective_X
== 9'd54 && effective_Y == 8'd22) || (effective_X == 9'd54 && effective_Y == 8'd23) ||
(effective_X == 9'd54 && effective_Y == 8'd24) || (effective_X == 9'd54 && effective_Y
== 8'd25) || (effective_X == 9'd54 && effective_Y == 8'd26) || (effective_X == 9'd54 &&
effective_Y == 8'd27) || (effective_X == 9'd54 && effective_Y == 8'd28) || (effective_X
== 9'd54 && effective_Y == 8'd29) || (effective_X == 9'd54 && effective_Y == 8'd30) ||
(effective_X == 9'd54 && effective_Y == 8'd31) || (effective_X == 9'd54 && effective_Y
== 8'd32) || (effective_X == 9'd54 && effective_Y == 8'd33) || (effective_X == 9'd54 &&
effective_Y == 8'd34) || (effective_X == 9'd54 && effective_Y == 8'd35) || (effective_X
== 9'd54 && effective_Y == 8'd36) || (effective_X == 9'd54 && effective_Y == 8'd37) ||
(effective_X == 9'd54 && effective_Y == 8'd38)

```

```

            ) begin

```

```

                pixel_colour = colourful ?

```

```

                (((randNum_12b[10:8] ^ randNum_12b[6:4]) == 3'b0) ? 3'b111 : (randNum_12b[10:8] ^
randNum_12b[6:4])) : 3'b111;

```

```

            end

```

```

        end

```

```

//            if (gameBoard_cur_Value == 12'd4096) begin

```

```

//                if(effective_X == 6'd1 && effective_Y == 6'd1 ||

```



[illegible]

```

== 8'd27) || (rand_eff_X == 9'd49 && rand_eff_Y == 8'd28) || (rand_eff_X == 9'd49 &&
rand_eff_Y == 8'd37) || (rand_eff_X == 9'd49 && rand_eff_Y == 8'd38) ||

                                (rand_eff_X == 9'd50 && rand_eff_Y == 8'd18) ||
(rand_eff_X == 9'd50 && rand_eff_Y == 8'd19) || (rand_eff_X == 9'd50 && rand_eff_Y
== 8'd27) || (rand_eff_X == 9'd50 && rand_eff_Y == 8'd28) || (rand_eff_X == 9'd50 &&
rand_eff_Y == 8'd37) || (rand_eff_X == 9'd50 && rand_eff_Y == 8'd38) ||

                                (rand_eff_X == 9'd51 && rand_eff_Y == 8'd18) ||
(rand_eff_X == 9'd51 && rand_eff_Y == 8'd19) || (rand_eff_X == 9'd51 && rand_eff_Y
== 8'd27) || (rand_eff_X == 9'd51 && rand_eff_Y == 8'd28) || (rand_eff_X == 9'd51 &&
rand_eff_Y == 8'd37) || (rand_eff_X == 9'd51 && rand_eff_Y == 8'd38) ||

                                (rand_eff_X == 9'd52 && rand_eff_Y == 8'd18) ||
(rand_eff_X == 9'd52 && rand_eff_Y == 8'd19) || (rand_eff_X == 9'd52 && rand_eff_Y
== 8'd27) || (rand_eff_X == 9'd52 && rand_eff_Y == 8'd28) || (rand_eff_X == 9'd52 &&
rand_eff_Y == 8'd37) || (rand_eff_X == 9'd52 && rand_eff_Y == 8'd38) ||

                                (rand_eff_X == 9'd53 && rand_eff_Y == 8'd18) ||
(rand_eff_X == 9'd53 && rand_eff_Y == 8'd19) || (rand_eff_X == 9'd53 && rand_eff_Y
== 8'd20) || (rand_eff_X == 9'd53 && rand_eff_Y == 8'd21) || (rand_eff_X == 9'd53 &&
rand_eff_Y == 8'd22) || (rand_eff_X == 9'd53 && rand_eff_Y == 8'd23) || (rand_eff_X ==
9'd53 && rand_eff_Y == 8'd24) || (rand_eff_X == 9'd53 && rand_eff_Y == 8'd25) ||
(rand_eff_X == 9'd53 && rand_eff_Y == 8'd26) || (rand_eff_X == 9'd53 && rand_eff_Y
== 8'd27) || (rand_eff_X == 9'd53 && rand_eff_Y == 8'd28) || (rand_eff_X == 9'd53 &&
rand_eff_Y == 8'd37) || (rand_eff_X == 9'd53 && rand_eff_Y == 8'd38) ||

                                (rand_eff_X == 9'd54 && rand_eff_Y == 8'd18) ||
(rand_eff_X == 9'd54 && rand_eff_Y == 8'd19) || (rand_eff_X == 9'd54 && rand_eff_Y
== 8'd20) || (rand_eff_X == 9'd54 && rand_eff_Y == 8'd21) || (rand_eff_X == 9'd54 &&
rand_eff_Y == 8'd22) || (rand_eff_X == 9'd54 && rand_eff_Y == 8'd23) || (rand_eff_X ==
9'd54 && rand_eff_Y == 8'd24) || (rand_eff_X == 9'd54 && rand_eff_Y == 8'd25) ||
(rand_eff_X == 9'd54 && rand_eff_Y == 8'd26) || (rand_eff_X == 9'd54 && rand_eff_Y
== 8'd27) || (rand_eff_X == 9'd54 && rand_eff_Y == 8'd28) || (rand_eff_X == 9'd54 &&
rand_eff_Y == 8'd37) || (rand_eff_X == 9'd54 && rand_eff_Y == 8'd38)

                                ) begin

                                rand_colour = colourful ? (((randNum_12b[2:0] ^
randNum_12b[9:7]) == 3'b0) ? 3'b111 : (randNum_12b[2:0] ^ randNum_12b[9:7])) :
3'b111;

                                end

//                                end

                                end

```

```

end

//  assign x = screen_X;
//  assign y = screen_Y;
//  assign colour = pixel_colour;

initial begin
    highscore <= 12'b0;
    randomNum_reg <= 4'b0;
    move_reg <= 4'b0;
    iteration_Counter <= 3'b0;
    casc_Counter <= 7'b0;
    clearBoard_Counter <= 5'b0;
    displayBoard_Counter <= 5'b0;
    rand_X <= 9'd0;
    rand_Y <= 8'd0;
    randNum_counter <= 7'b0;
    sig_randNum_GOOD = 1'b0;
    sig_doneProcess = 1'b0;
    sig_toNoMove = 1'b0;
    sig_toMergeMove = 1'b0;
    sig_toJustMove = 1'b0;
    sig_nextIteration = 1'b0;
    temp_X = 2'b0;
    temp_Y = 2'b0;
    temp_iter_counter = 3'b0;
    sig_drawBoard_Cont = 1'b1;
    sig_doneCasc = 1'b0;

```

```

    temp_casc_Counter = 7'b0;
    temp_highscore = 12'b0;
    sig_gameLose = 1'b0;
    gameRAM_Addr = 4'b0;
    gameRAM_DataIn = 12'd0;
    sig_clearBoard_DONE = 1'b0;
    sig_drawBoard_DONE = 1'b0;
    sig_debug_displayBoard_DONE = 1'b0;
    sig_randNumDraw_DONE = 1'b0;
    screen_X <= 9'd0;
    screen_Y <= 8'd0;
    pixel_colour = 3'b000;
    rand_colour = 3'b000;
end

endmodule

```

```

module Linear_FB_Shift_Reg_5b(
    input CLOCK_50,
    input resetn,
    input LFBSR_enable,
    output reg [4:0] out
);

    wire feedback;

    assign feedback = ~(out[4] ^ out[2]);

```

```

always @(posedge CLOCK_50, negedge resetn) begin
    if (!resetn)
        out = 5'b0;
    else if (LFBSR_enable) begin
        out = {out[3:0],feedback};
    end
end

endmodule

```

```

module Linear_FB_Shift_Reg_12b(
    input CLOCK_50,
    input resetn,
    input LFBSR_enable,
    output reg [11:0] out
);

    wire feedback;

    assign feedback = ~(out[11] ^ out[8] ^ out[7] ^ out[4]);

    always @(posedge CLOCK_50, negedge resetn) begin
        if (!resetn)
            out = 12'b101010101010;
        else if (LFBSR_enable) begin
            out = {out[10:0],feedback};
        end
    end
end

```



```
endmodule
```

```
module counter_4b(  
    input CLOCK_50,  
    input resetn,  
    input counter_4b_enable,  
    output reg [3:0] out  
);  
  
    always @(posedge CLOCK_50, negedge resetn) begin  
        if (!resetn)  
            out = 4'b0;  
        else if (counter_4b_enable) begin  
            out = out + 1'b1;  
        end  
    end  
end
```

```
endmodule
```

```
module hex_decoder(hex_digit, segments);  
    input [3:0] hex_digit;  
    output reg [6:0] segments;  
  
    always @(*)  
        case (hex_digit)  
            4'h0: segments = 7'b100_0000;  
            4'h1: segments = 7'b111_1001;
```

```

    4'h2: segments = 7'b010_0100;
    4'h3: segments = 7'b011_0000;
    4'h4: segments = 7'b001_1001;
    4'h5: segments = 7'b001_0010;
    4'h6: segments = 7'b000_0010;
    4'h7: segments = 7'b111_1000;
    4'h8: segments = 7'b000_0000;
    4'h9: segments = 7'b001_1000;
    4'hA: segments = 7'b000_1000;
    4'hB: segments = 7'b000_0011;
    4'hC: segments = 7'b100_0110;
    4'hD: segments = 7'b010_0001;
    4'hE: segments = 7'b000_0110;
    4'hF: segments = 7'b000_1110;
    default: segments = 7'h7f;
endcase
endmodule

```

```

module keyboard_press_driver(
    input  CLOCK_50,
    output reg valid, makeBreak,
    output reg [7:0] outCode,
    output reg [3:0] sig_move,
    output reg KEYBOARD_ENTER,
    output reg KEYBOARD_RESET,
    input  PS2_DAT, // PS2 data line
    input  PS2_CLK, // PS2 clock line
    input reset

```

);

```
parameter FIRST = 1'b0, SEENF0 = 1'b1;
reg state;
reg [1:0] count;
wire [7:0] scan_code;
reg [7:0] filter_scan;
wire scan_ready;
reg read;
parameter NULL = 8'h00;

wire [7:0] ARROW_UP = 8'h75;  //codes for arrows
wire [7:0] ARROW_DOWN = 8'h72;
wire [7:0] ARROW_LEFT = 8'h6B;
wire [7:0] ARROW_RIGHT = 8'h74;
wire [7:0] SPACEBAR = 8'h29;
wire [7:0] ESC = 8'h76;
wire [7:0] ENTER = 8'h5A;

initial
begin
    state = FIRST;
    filter_scan = NULL;
    read = 1'b0;
    count = 2'b00;
end
```

```

// inner driver that handles the PS2 keyboard protocol
// outputs a scan_ready signal accompanied with a new scan_code
keyboard_inner_driver kbd(
    .keyboard_clk(PS2_CLK),
    .keyboard_data(PS2_DAT),
    .clock50(CLOCK_50),
    .reset(reset),
    .read(read),
    .scan_ready(scan_ready),
    .scan_code(scan_code)
);

always @(posedge CLOCK_50) begin
    case(count)
        2'b00:
            if(scan_ready)
                count <= 2'b01;
        2'b01:
            if(scan_ready)
                count <= 2'b10;
        2'b10:
            begin
                read <= 1'b1;
                count <= 2'b11;
                valid <= 0;
                outCode <= scan_code;
                case(state)
                    FIRST:

```

```

case(scan_code)
    8'hF0:
        begin
            state <= SEENF0;
        end
    8'hE0:
        begin
            state <= FIRST;
        end
    default:
        begin
            filter_scan <=
scan_code;
            if(filter_scan !=
scan_code)
                begin
                    valid
<= 1'b1;
                    makeBreak <= 1'b1;
                end
            end
        end
    endcase
SEENF0:
    begin
        state <= FIRST;
        if(filter_scan == scan_code)
            begin
                filter_scan <=
NULL;

```

```

                                end
                                valid <= 1'b1;
                                makeBreak <= 1'b0;
                                end
                                endcase
                                end
                                2'b11:
                                begin
                                    read <= 1'b0;
                                    count <= 2'b00;
                                    valid <= 0;
                                end
                                endcase
                                end

```

```

always @ (posedge CLOCK_50) begin
    sig_move <= 4'b0;
    KEYBOARD_ENTER <= 1'b0;
    KEYBOARD_RESET <= 1'b0;
    if (outCode == ARROW_UP)
        sig_move[2] <= 1'b1 & makeBreak;
    else if (outCode == ARROW_DOWN)
        sig_move[1] <= 1'b1 & makeBreak;
    else if (outCode == ARROW_LEFT)
        sig_move[3] <= 1'b1 & makeBreak;
    else if (outCode == ARROW_RIGHT)
        sig_move[0] <= 1'b1 & makeBreak;
    else if (outCode == ESC)

```

```

        KEYBOARD_RESET <= 1'b1 & makeBreak;
    else if (outCode == SPACEBAR)
        KEYBOARD_RESET <= 1'b1 & makeBreak;
    else if (outCode == ENTER)
        KEYBOARD_ENTER <= 1'b1 & makeBreak;
    end

endmodule

module keyboard_inner_driver(keyboard_clk, keyboard_data, clock50, reset, read,
scan_ready, scan_code);
    input keyboard_clk;
    input keyboard_data;
    input clock50; // 50 Mhz system clock
    input reset;
    input read;
    output scan_ready;
    output [7:0] scan_code;
    reg ready_set;
    reg [7:0] scan_code;
    reg scan_ready;
    reg read_char;
    reg clock; // 25 Mhz internal clock

    reg [3:0] incnt;
    reg [8:0] shiftin;

    reg [7:0] filter;
    reg keyboard_clk_filtered;

```

```

// scan_ready is set to 1 when scan_code is available.
// user should set read to 1 and then to 0 to clear scan_ready

always @ (posedge ready_set or posedge read)
if (read == 1) scan_ready <= 0;
else scan_ready <= 1;

// divide-by-two 50MHz to 25MHz
always @(posedge clock50)
    clock <= ~clock;

// This process filters the raw clock signal coming from the keyboard
// using an eight-bit shift register and two AND gates

always @(posedge clock)
begin
    filter <= {keyboard_clk, filter[7:1]};
    if (filter==8'b1111_1111) keyboard_clk_filtered <= 1;
    else if (filter==8'b0000_0000) keyboard_clk_filtered <= 0;
end

// This process reads in serial data coming from the terminal

always @(posedge keyboard_clk_filtered)

```



```

begin
    if (reset==1)
    begin
        incnt <= 4'b0000;
        read_char <= 0;
    end
    else if (keyboard_data==0 && read_char==0)
    begin
        read_char <= 1;
        ready_set <= 0;
    end
    else
    begin
        // shift in next 8 data bits to assemble a scan code
        if (read_char == 1)
        begin
            if (incnt < 9)
            begin
                incnt <= incnt + 1'b1;
                shiftin = { keyboard_data, shiftin[8:1]};
                ready_set <= 0;
            end
        else
        begin
            incnt <= 0;
            scan_code <= shiftin[7:0];
            read_char <= 0;
            ready_set <= 1;
        end
    end
end

```

```
end  
end  
end  
end  
endmodule
```