Verilog HDL高级数字设计

实验报告

题目：“俄罗斯方块”FPGA实现

## 实验目的

通过此次项目，完成以下目的：

1. 熟悉Xilinx FPGA的架构及开发流程
2. 设计一个功能完整的系统，掌握FSM + Datapath的设计方法。

## 实验内容

1. 项目介绍

本项目主要在FPGA上实现了一个经典小游戏“俄罗斯方块”。本项目基本解决方案是，使用Xilinx Zynq系列开发板ZedBoard作为平台，实现主控模块，通过VGA接口来控制屏幕进行显示。

1. 系统框架

整个系统由四部分组成，按键输入处理模块、控制模块、数据路径模块以及VGA显示接口模块。整个系统的结构如下图所示：



图1：系统框图

下面分别对四个模块进行介绍：

1. 按键输入处理模块

按键处理模块的主要功能是对输入系统的up，down，left，right四个控制信号进行消抖处理，并对其进行上升沿检测。

消抖模块采用上课所提出的结构，采用了一个4位的移位寄存器，先将输入信号延迟4个时钟周期，再对其以一个较低的时钟频率进行采用。消抖模块的结构如下图所示：

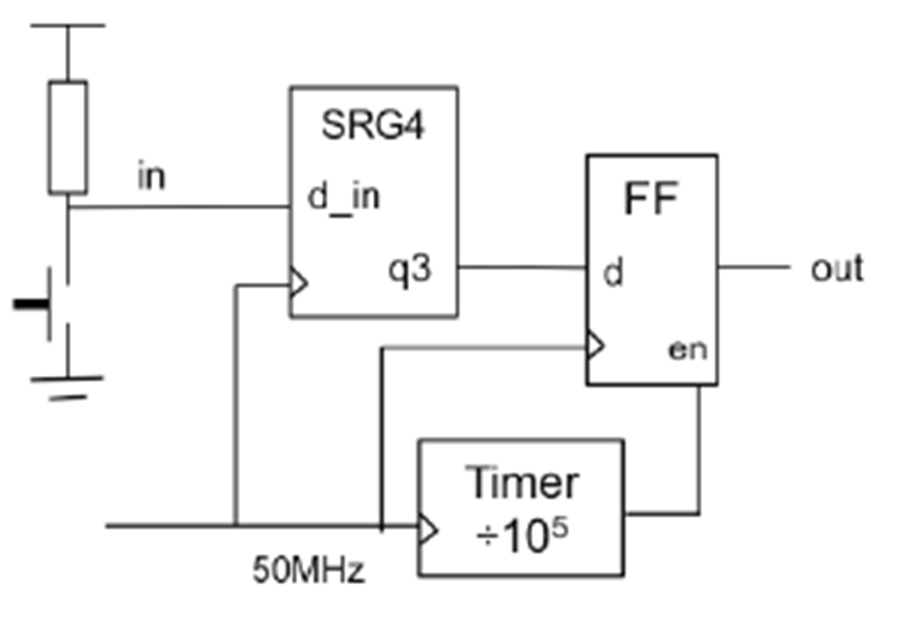


图2：消抖模块结构示意图

为了简化控制系统，在本系统的设计过程中，不考虑长时间按键产生连按效果。因而，需要对按键进行上升沿检测。上升沿检测的基本实现方案是加入一组寄存器，对前一个的按键信号进行暂存，将暂存的值与当前值进行比较，当上一个值为0而当前值为1时，即认为其检测到了一个上升沿。

1. 控制模块

控制模块采用FSM的方式进行控制。在控制模块中，定义了10个状态：

S\_idle：上电复位后进入的空状态，当start信号为1时进入S\_new状态

S\_new：用于产生新的俄罗斯方块。

S\_hold：保持状态。在这个状态中进行计时，当时间到达一定间隔时，转到S\_down状态；或者等待输入信号（up，down，left，right）时，转到S\_down（按键为down）或者S\_move（up，left，right）状态。

S\_down：判断当前俄罗斯块能否下移一格。如果可以，则转到S\_remove\_1状态，如果不行，则转到S\_shift状态。

S\_move：判断当前俄罗斯块能够按照按键信号指定的指令进行移动，如果可以，则转到S\_shift状态，如果不可以，则转到S\_remove\_1状态。

S\_shift：更新俄罗斯方块的坐标信息。返回S\_hold。

S\_remove\_1：更新整个屏幕的矩阵信息。转移到S\_remove\_2状态。

S\_remove\_2：判断是否可以消除，将可以消除的行消除，并将上面的行下移一行。重复此过程，直到没有可消除的行为止。跳转到S\_isdie状态

S\_isdie：判断是否游戏结束。如果结束，则跳转到S\_stop状态。如果没有，则跳转到S\_new状态，生成新的俄罗斯方块。

S\_stop：清楚整个屏幕，并跳转到S\_idle状态。

整个控制过程的ASMD图如下图所示：



图3： 控制模块ASMD图

1. 数据路径

数据路径模块主要功能是，根据控制模块给出的信号，对俄罗斯方块当前的逻辑状态进行判断，更新背景矩阵。具体如下：

方块：

方块分为非活动方块与活动方块。非活动方块为：（1）之前下落的方块；（2）下落后方块消除之后的结果。由背景矩阵表示。活动方块为当前下落中的方块，由活动方块坐标与方块类型表示（后简称方块）。

背景矩阵：

reg [9:0] R [23:0];

背景矩阵R是24行10列的寄存器组，负责保存非活动方块坐标，即R中任一位置，如方块存在，则该位置1，否则为0。

活动方块坐标：

output reg [4:0] n,

output reg [3:0] m,

n, m分别为当前活动方块的行、列指针，指向方块固定点位置。方块固定点为方块旋转时不变的格点，依据方块种类决定，下文方块模型中详述。

方块类型：

output reg [6:0] BLOCK,

BLOCK代表方块类型，由7位编码构成。

数据交换：

Datapath与其余模块的数据交换分为两部分：（1）与control\_unit间的状态指令交互；（2）控制merge，间接实现对VGA的控制。

方块模型：

俄罗斯方块共有7中形状的方块（O,L,J,I,T,Z,S），每种方块有1-4种不同的旋转变形方式。为方便起见，将方块定位A-G，旋转编号为1-4，将方块编码成A\_1-G\_2的19种，如 下图：

图中，深色方块是该种方块的固定点。

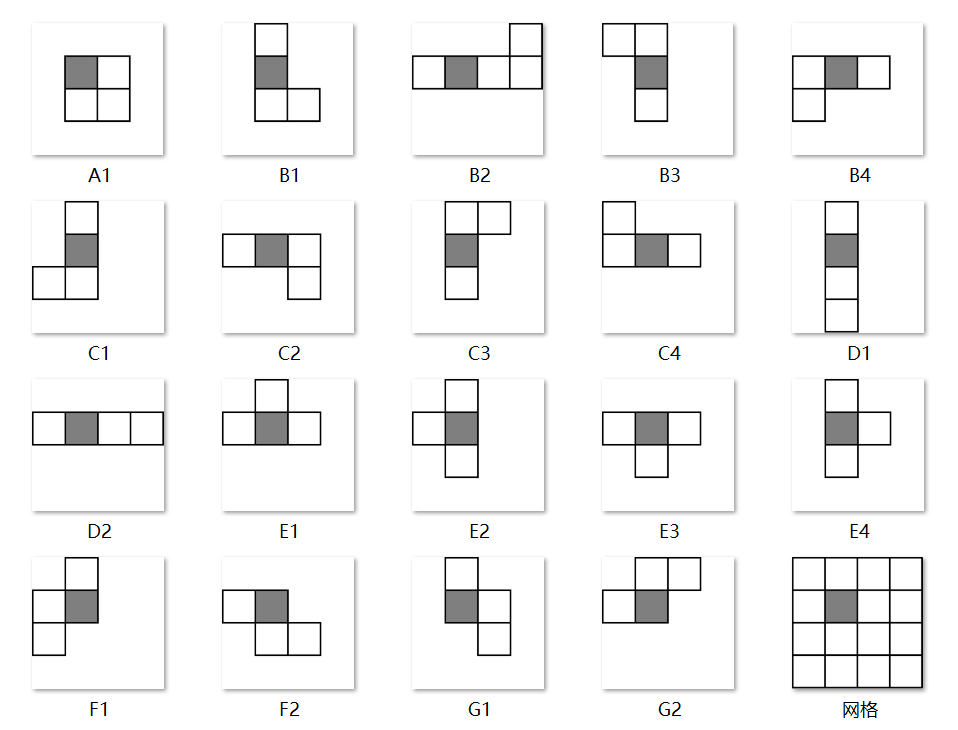


图4： 方块模型示意图

方块运动：

产生：

方块产生由一个简单的伪随机过程决定。系统采用一个3位的计数器产生随机数，进入S\_new，BLOCK的值被NEW\_BLOCK覆盖，方块坐标n<=1;m<=5;同时，根据计数器，NEW\_BLOCK的值刷新为A\_1,B\_1,…,G\_1中的一种，作为下一次方块。

移动：

方块移动分为四种：旋转，下落，向左，向右，由键盘KEYBOARD=[UP, DOWN, LEFT, RIGHT]控制。移动分两步进行：（1）判断；（2）转换。

判断过程包含S\_down,S\_move。判断分两步：首先，判断变换后方块坐标是否合法，即变换后是否会造成方块越界。然后，判断变换后方块可能占据的新位置是否有背景矩阵方块存在。两步判断通过后返回成功信号，否则失败。因判断代码量较多，仅举一例说明：

判断D\_1向右运动（MOVE\_ABLE初值为0）：

if (m<=8)

if (!((R[n-1][m+1])|(R[n][m+1])|(R[n+1][m+1])|(R[n+2][m+1])))

MOVE\_ABLE=1;

else MOVE\_ABLE=0;

转换过程（S\_shift）进行方块的移动或变形。根据KEYBOARD，移动时，改变方块坐标；变形时，方块按类别变换，如：A\_1→A\_1;B\_1→B\_2; B\_2→B\_3; B\_4→B\_1;

停止与消除：

方块停止与消除由两个状态完成：S\_remove1,S\_remove2。

前一状态中，根据BLOCK, n, m，将活动方块位置覆盖至R，变为非活动方块。

后一状态中，根据行满状态，进行行的消除与平移,具体如下：

显然，俄罗斯方块能影响的最大行数为4，因此，在REMOVE\_2中，仅对R[n-1],R[n],R[n+1],R[n+2]四行依次进行处理。处理过程为：如果该行(k)满，则由k行开始，至1行结束，逐行向下平移，当前平移位置由计数器REMOVE\_2\_C控制，当前行消除截止由标志位SIG确认。

每行处理完后，将REMOVE\_FINISH[3:0]中相应位置1，REMOVE\_FINISH全1时，REMOVE\_2完成。

死亡判定：

R中的0-3行位于屏幕上方，不进行显示，仅有新生成的方块坐标会进入这一区域。因而，当消除完成后，如R[3]不为空，游戏结束。

1. 显示部分

输出结果通过VGA接口接入显示屏显示。VGA（Video Graphics Array）视频图形阵列是IBM于1987年提出的一个使用模拟信号的电脑显示标准。VGA接口即电脑采用VGA标准输出数据的专用接口。VGA接口共有15针，分成3排，每排5个孔，显卡上应用最为广泛的接口类型，绝大多数显卡都带有此种接口。它传输红、绿、蓝模拟信号以及同步信号(水平和垂直信号)。

使用Verilog HDL语言对VGA进行控制一般只需控制行扫描信号、列扫描信号和红绿蓝三色信号输出即可。

VGA输出可分为四个模块：时钟分频模块、数据组织模块、接口控制模块和顶层模块。以下进行分块描述。

时钟模块分频模块对FPGA系统时钟进行分频。由于使用的显示屏参数为640\*480\*60Hz，其真实屏幕大小为800\*525，因此所需时钟频率为800\*525\*60Hz=25.175MHz，可近似处理为25MHz。FPGA系统时钟为100M，因此将其四分频即可基本满足显示要求。

数据组织模块是将预备输出的数据组织为可以通过VGA接口控制的数据形式，本次设计中因接口已经协调，数据可不经过此模块进行组织，故可忽略该模块。

接口控制模块通过VGA接口对显示屏进行控制。VGA的扫描顺序是从左到右，从上到下。例如在640X480的显示模式下，从显示器的左上角开始往右扫描，直到640个像素扫完，再回到最左边，开始第二行的扫描，如此往复，到第480行扫完时即完成一帧图像的显示。这时又回到左上角，开始下一帧图像的扫描。如果每秒能完成60帧，则称屏幕刷新频率为60Hz。宏观上，一帧屏幕由480个行和640个列填充而成，而实际上，一帧屏幕除了显示区，还包含其他未显示部分，作为边框或者用来同步。具体而言，一个完整的行同步信号包含了左边框、显示区、右边框还有返回区四个部分，总共800个像素，其分配如下：

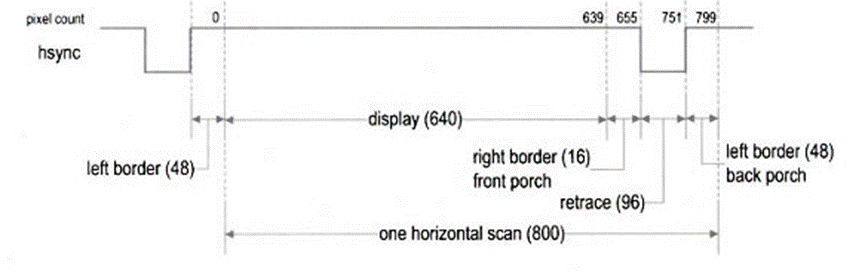


图5： VGA行扫描时序

同样的，一个完整的垂直同步信号也分为四个区域，总共525个像素，分配如下：

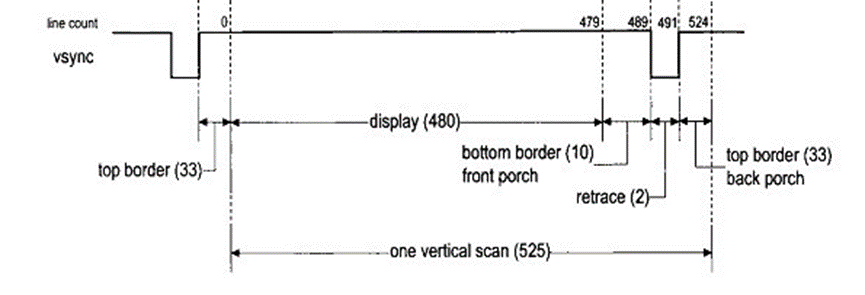


图6：VGA场扫描时序

模块通过组织输出行扫描信号、列扫描信号和三原色信号对显示屏实现控制。

## 实验结果

实验结果图如下：

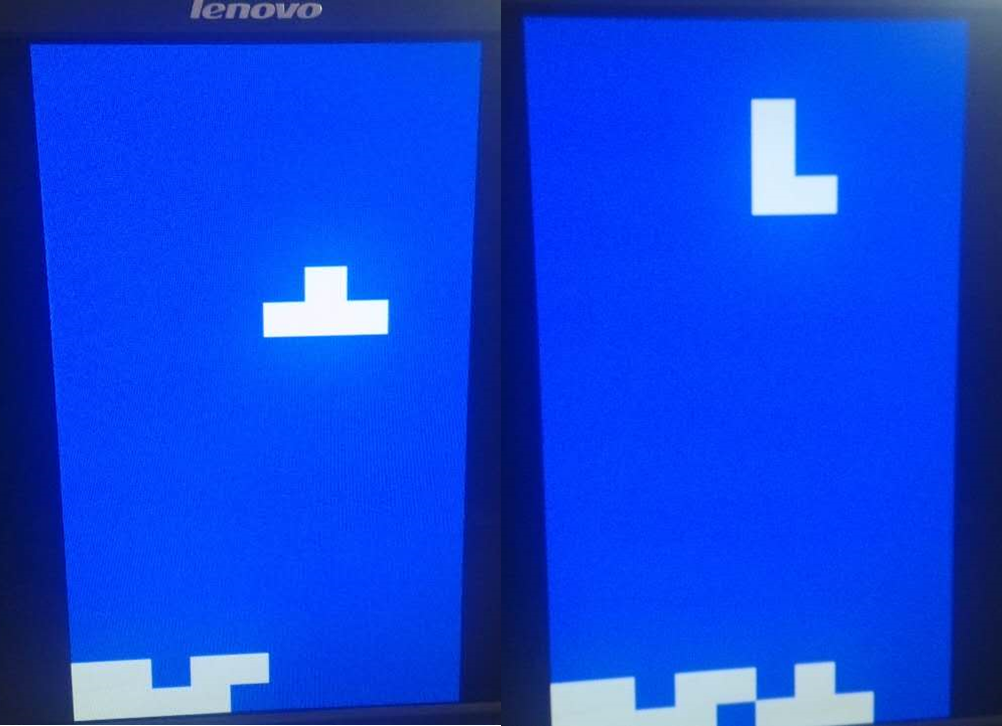


图7：实验结果图

## 实验总结

* 1. 完成情况

本次项目我们完成了既定目标，即完成一个经典小游戏“俄罗斯方块”的核心功能。在本次实验过程中，我们通过采用分工合作的方式，通过对系统功能的分析，确定解决方案，完成了对一个系统自上而下的设计，并尝试采用控制单元+数据路径这样的方式来处理核心模块。

* 1. 不足与改进之处

由于时间仓促，加之对俄罗斯方块逻辑复杂度估计不足，到最后展示之前我们才完成了对核心模块的调试。因此，在用户界面上没有做过多的调整。另外，由于在进行模块划分时，一些接口没有事先定义好，导致在最后系统整合时，不得不进行修改与调整，由此而造成了一部分时间的浪费。

总的来说，通过这个项目，小组成员对于硬件设计“并行”的特点有了比较直接的认识，同时也在调试的过程中掌握了一些硬件调试常用的方法，也认识到了仿真的重要意义所在。另外就是关于团队协作方面的一个教训，在系统划分时要注意把接口定义好，以免造成不必要的代价。

## 实验代码

KeyBoard模块

`timescale 1ns **/** 1ps

**module** key**(**

**input** clk**,**

**input** rst\_n**,**

**input** UP\_KEY**,**

**input** LEFT\_KEY**,**

**input** RIGHT\_KEY**,**

**input** DOWN\_KEY**,**

**output** **reg** rotate**,**

**output** **reg** left**,**

**output** **reg** right**,**

**output** **reg** down

**);**

**reg** **[**3**:**0**]** shift\_up**;**

**reg** **[**3**:**0**]** shift\_left**;**

**reg** **[**3**:**0**]** shift\_right**;**

**reg** **[**3**:**0**]** shift\_down**;**

**always** **@(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

shift\_up **<=** 0**;**

**else**

shift\_up **<=** **{**shift\_up**[**2**:**0**],** UP\_KEY**};**

**end**

**always** **@(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

shift\_right **<=** 0**;**

**else**

shift\_right **<=** **{**shift\_right**[**2**:**0**],** RIGHT\_KEY**};**

**end**

**always** **@(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

shift\_left **<=** 0**;**

**else**

shift\_left **<=** **{**shift\_left**[**2**:**0**],** LEFT\_KEY**};**

**end**

**always** **@(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

shift\_down **<=** 0**;**

**else**

shift\_down **<=** **{**shift\_down**[**2**:**0**],** DOWN\_KEY**};**

**end**

**reg** clk\_div**;**

**reg** **[**7**:**0**]** clk\_cnt**;**

**always** **@** **(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

**begin**

clk\_cnt **<=** 0**;**

clk\_div **<=** 0**;**

**end**

**else** **if** **(**clk\_cnt **<=** 8'd49**)**

**begin**

clk\_cnt **<=** clk\_cnt **+** 1**;**

clk\_div **<=** clk\_div**;**

**end**

**else**

**begin**

clk\_cnt **<=** 0**;**

clk\_div **<=** **~**clk\_div**;**

**end**

**end**

**always** **@(posedge** clk\_div **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

**begin**

rotate **<=** 0**;**

left **<=** 0**;**

right **<=** 0**;**

down **<=** 0**;**

**end**

**else**

**begin**

rotate **<=** shift\_up**[**3**];**

left **<=** shift\_left**[**3**];**

right **<=** shift\_right**[**3**];**

down **<=** shift\_down**[**3**];**

**end**

**end**

**endmodule**

控制模块程序

**module** game\_control\_unit **(**

**input** clk**,**

**input** rst\_n**,**

**input** rotate**,**

**input** left**,**

**input** right**,**

**input** down**,**

**input** start**,**

**output** **reg** **[**3**:**0**]** opcode**,**

**output** **reg** gen\_random**,**

**output** **reg** hold**,**

**output** **reg** shift**,**

**output** **reg** move\_down**,**

**output** **reg** remove\_1**,**

**output** **reg** remove\_2**,**

**output** **reg** stop**,**

**output** **reg** move**,**

**output** **reg** isdie**,**

**output** **reg** auto\_down**,**

**input** shift\_finish**,**

**input** remove\_2\_finish**,**

**input** down\_comp**,**

**input** move\_comp**,**

**input** die

**);**

**reg** left\_reg**;**

**reg** right\_reg**;**

**reg** up\_reg**;**

**reg** down\_reg**;**

**always** **@(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

**begin**

left\_reg **<=** 0**;**

right\_reg **<=** 0**;**

up\_reg **<=** 0**;**

down\_reg **<=** 0**;**

**end**

**else**

**begin**

left\_reg **<=** left**;**

right\_reg **<=** right**;**

up\_reg **<=** rotate**;**

down\_reg **<=** down**;**

**end**

**end**

**reg** auto\_down\_reg**;**

**always** **@** **(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

auto\_down\_reg **<=** 0**;**

**else** **if** **(**time\_cnt **==** time\_val**)**

auto\_down\_reg **<=** 1**;**

**else**

auto\_down\_reg **<=** 0**;**

**end**

**always** **@** **(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

auto\_down **<=** 0**;**

**else**

auto\_down **<=** auto\_down\_reg**;**

**end**

**parameter** time\_val **=** 26'd25000001**;**

**reg** **[**25**:**0**]** time\_cnt**;**

**localparam** S\_idle **=** 4'd0**,**

S\_new **=** 4'd1**,**

S\_hold **=** 4'd2**,**

S\_move **=** 4'd3**,**

S\_shift **=** 4'd4**,**

S\_down **=** 4'd5**,**

S\_remove\_1 **=** 4'd6**,**

S\_remove\_2 **=** 4'd7**,**

S\_isdie **=** 4'd8**,**

S\_stop **=** 4'd9**;**

**reg** **[**3**:**0**]** state**,** next\_state**;**

**always** **@(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

state **<=** S\_idle**;**

**else**

state **<=** next\_state**;**

**end**

**always** **@** **(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

time\_cnt **<=** 0**;**

**else** **if** **(**hold **==** 0 **&&** time\_cnt **<** time\_val**)**

time\_cnt **<=** time\_cnt **+** 1**;**

**else** **if** **(**move\_down **==** 1**)**

time\_cnt **<=** 0**;**

**else** **begin**

time\_cnt **<=** time\_cnt**;**

**end**

**end**

**always** **@** **(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)** opcode**<=**0**;**

**else** opcode**<={**right**,** left**,** down**,** rotate**};**

**end**

**always** **@** **(\*)**

**begin**

next\_state **=** S\_idle**;**

hold **=** 1**;**

gen\_random **=** 0**;**

//opcode = 4'b0000;

shift **=** 0**;**

move\_down **=** 0**;**

remove\_1 **=** 0**;**

remove\_2 **=** 0**;**

stop **=** 0**;**

move **=** 0**;**

isdie **=** 0**;**

**case** **(**state**)**

S\_idle**:**

**begin**

**if** **(**start**)**

next\_state **=** S\_new**;**

**else**

next\_state **=** S\_idle**;**

**end**

S\_new**:**

**begin**

gen\_random **=** 1**;**

next\_state **=** S\_hold**;**

**end**

S\_hold**:**

**begin**

hold **=** 0**;**

**if** **(**time\_cnt **==** time\_val**)**

**begin**

next\_state **=** S\_down**;**

**end**

**else** **if** **((**down\_reg **==** 0**)** **&&** **(**down **==** 1**))**

**begin**

next\_state **=** S\_down**;**

**end**

**else** **if** **((**left\_reg **==** 0 **&&** left **==** 1**)||** **(** right\_reg **==** 0 **&&** right **==** 1**)||(**up\_reg **==** 0 **&&** rotate **==** 1**))**

**begin**

next\_state **=** S\_move**;**

**end**

**else**

next\_state **=** S\_hold**;**

**end**

S\_move**:**

**begin**

move **=** 1**;**

**if** **(**move\_comp**)**

next\_state **=** S\_shift**;**

**else**

next\_state **=** S\_hold**;**

**end**

S\_shift**:**

**begin**

shift **=** 1**;**

next\_state **=** S\_hold**;**

**end**

S\_down**:**

**begin**

move\_down **=** 1**;**

**if** **(**down\_comp**)**

next\_state **=** S\_shift**;**

**else**

next\_state **=** S\_remove\_1**;**

**end**

S\_remove\_1**:**

**begin**

remove\_1 **=** 1**;**

next\_state **=** S\_remove\_2**;**

**end**

S\_remove\_2**:**

**begin**

remove\_2 **=** 1**;**

**if** **(**remove\_2\_finish**)**

next\_state **=** S\_isdie**;**

**else**

next\_state **=** S\_remove\_2**;**

**end**

S\_isdie**:**

**begin**

isdie **=** 1**;**

**if** **(**die **==** 1**)**

next\_state **=** S\_stop**;**

**else**

next\_state **=** S\_new**;**

**end**

S\_stop**:**

**begin**

stop **=** 1**;**

next\_state **=** S\_idle**;**

**end**

**default** next\_state **=** S\_idle**;**

**endcase**

**end**

**endmodule**

数据路径

**module** Datapath\_Unit **#(**

**parameter** A\_1 **=** 7'b0001000**,**

B\_1 **=** 7'b0011000**,**

B\_2 **=** 7'b0010100**,**

B\_3 **=** 7'b0010010**,**

B\_4 **=** 7'b0010001**,**

C\_1 **=** 7'b0101000**,**

C\_2 **=** 7'b0100100**,**

C\_3 **=** 7'b0100010**,**

C\_4 **=** 7'b0100001**,**

D\_1 **=** 7'b0111000**,**

D\_2 **=** 7'b0110100**,**

E\_1 **=** 7'b1001000**,**

E\_2 **=** 7'b1000100**,**

E\_3 **=** 7'b1000010**,**

E\_4 **=** 7'b1000001**,**

F\_1 **=** 7'b1011000**,**

F\_2 **=** 7'b1010100**,**

G\_1 **=** 7'b1101000**,**

G\_2 **=** 7'b1100100

**)(**

**output** **reg** MOVE\_ABLE**,**SHIFT\_FINISH**,**DOWN\_ABLE**,**DIE\_TRUE**,**

**output** **[**239**:**0**]** M\_OUT**,**

**output** **reg** **[**4**:**0**]** n**,**

**output** **reg** **[**3**:**0**]** m**,**

**output** **reg** **[**6**:**0**]** BLOCK**,**

//output reg REMOVE\_1\_FINISH,

**output** **reg** REMOVE\_2\_FINISH**,**

//output reg NEW\_BLOCK,

**input** clk**,**rst\_n**,**MOVE**,**DOWN**,**DIE**,**SHIFT**,**REMOVE\_1**,**REMOVE\_2**,**NEW**,**STOP**,**AUTODOWN**,**

**input** **[**3**:**0**]** KEYBOARD

**);**

**reg** **[**2**:**0**]** RAN**;**

**reg** **[**9**:**0**]** R **[**23**:**0**];**

**reg** **[**6**:**0**]** NEW\_BLOCK**;**

**reg** **[**6**:**0**]** BLOCK\_P**;**

**reg** **[**4**:**0**]** remove\_cnt**;**

**reg** **[**3**:**0**]** REMOVE\_2\_S**;**

**reg** **[**3**:**0**]** REMOVE\_FINISH**;**

**reg** **[**4**:**0**]** REMOVE\_2\_C**;**

**reg** SIG**;**

**always** **@** **(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

RAN**<=**0**;**

**else** **if** **(**RAN**==**7**)** RAN**<=**1**;**

**else** RAN**<=**RAN**+**1**;**

**end**

// MOVE\_ABLE signal

**always** **@** **(\*)**

**begin**

MOVE\_ABLE **=** 0**;**

**if** **(**MOVE**)**

**begin**

**if** **(**KEYBOARD**[**0**])** //UP

**begin**

// MOVE\_ABLE=1;

**case** **(**BLOCK**)**

A\_1**:** MOVE\_ABLE**=**0**;**

B\_1**:** **if** **(**m**>=**1**)**

**begin** **if** **(!((**R**[**n**][**m**-**1**])|(**R**[**n**][**m**+**1**])|(**R**[**n**-**1**][**m**+**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

B\_2**:** **if** **(**n**<=**22**)**

**begin** **if** **(!((**R**[**n**+**1**][**m**-**1**])|(**R**[**n**+**1**][**m**])|(**R**[**n**-**1**][**m**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

B\_3**:** **if** **(**m**<=**8**)**

**begin** **if** **(!(**R**[**n**][**m**-**1**]** **|** R**[**n**][**m**+**1**]** **|** R**[**n**+**1**][**m**-**1**]))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;** **end**

B\_4**:** **begin** **if** **(!((**R**[**n**-**1**][**m**])|(**R**[**n**+**1**][**m**])|(**R**[**n**+**1**][**m**+**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

C\_1**:** **if** **(**m**<=**8**)**

**begin** **if** **(!((**R**[**n**][**m**-**1**])|(**R**[**n**][**m**+**1**])|(**R**[**n**+**1**][**m**+**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

C\_2**:** **begin** **if** **(!((**R**[**n**-**1**][**m**])|(**R**[**n**-**1**][**m**+**1**])|(**R**[**n**-**1**][**m**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

C\_3**:** **if** **(**m**>=**1**)**

**begin** **if** **(!((**R**[**n**-**1**][**m**-**1**])|(**R**[**n**][**m**-**1**])|(**R**[**n**][**m**+**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

C\_4**:** **if** **(**n**<=**22**)**

**begin** **if** **(!((**R**[**n**-**1**][**m**])|(**R**[**n**+**1**][**m**-**1**])|(**R**[**n**+**1**][**m**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

D\_1**:** **if** **((**m**>=**1**)&(**m**<=**7**))**

**begin** **if** **(!((**R**[**n**][**m**-**1**])|(**R**[**n**][**m**+**1**])|(**R**[**n**][**m**+**2**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

D\_2**:** **if** **(**n**<=**21**)**

**begin** **if** **(!((**R**[**n**-**1**][**m**])|(**R**[**n**+**1**][**m**])|(**R**[**n**+**2**][**m**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

E\_1**:** **if** **(**n**<=**22**)**

**begin** **if** **(!(**R**[**n**+**1**][**m**]))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

E\_2**:** **if** **(**m**<=**8**)**

**begin** **if** **(!(**R**[**n**][**m**+**1**]))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

E\_3**:** **begin** **if** **(!(**R**[**n**-**1**][**m**]))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

E\_4**:** **if** **(**m**>=**1**)**

**begin** **if** **(!(**R**[**n**][**m**-**1**]))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

F\_1**:** **if** **(**m**>=**1**)**

**begin** **if** **(!((**R**[**n**-**1**][**m**-**1**])|(**R**[**n**-**1**][**m**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

F\_2**:** **if** **(**n**<=**22**)**

**begin** **if** **(!((**R**[**n**-**1**][**m**+**1**])|(**R**[**n**+**1**][**m**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

G\_1**:** **if** **(**m**>=**1**)**

**begin** **if** **(!((**R**[**n**-**1**][**m**+**1**])|(**R**[**n**][**m**-**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

G\_2**:** **if** **(**n**<=**22**)**

**begin** **if** **(!((**R**[**n**][**m**+**1**])|(**R**[**n**+**1**][**m**+**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;end**

**default** MOVE\_ABLE**=**0**;**

**endcase**

**end**

**else** **if** **(**KEYBOARD**[**2**])** //LEFT

**begin**

// MOVE\_ABLE<=0;

**case** **(**BLOCK**)**

A\_1**:** **if** **(**m**>=**1**)** **if** **(!((**R**[**n**+**1**][**m**-**1**])|(**R**[**n**][**m**-**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

B\_1**:** **if** **(**m**>=**1**)** **if** **(!((**R**[**n**-**1**][**m**-**1**])|(**R**[**n**][**m**-**1**])|(**R**[**n**+**1**][**m**-**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

B\_2**:** **if** **(**m**>=**2**)** **if** **(!((**R**[**n**][**m**-**1**])|(**R**[**n**-**1**][**m**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

B\_3**:** **if** **(**m**>=**2**)** **if** **(!((**R**[**n**-**1**][**m**-**2**])|(**R**[**n**][**m**-**1**])|(**R**[**n**+**1**][**m**-**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

B\_4**:** **if** **(**m**>=**2**)** **if** **(!((**R**[**n**][**m**-**2**])|(**R**[**n**+**1**][**m**-**2**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

C\_1**:** **if** **(**m**>=**2**)** **if** **(!((**R**[**n**-**1**][**m**-**1**])|(**R**[**n**][**m**-**1**])|(**R**[**n**+**1**][**m**-**2**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

C\_2**:** **if** **(**m**>=**2**)** **if** **(!((**R**[**n**][**m**-**2**])|(**R**[**n**+**1**][**m**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

C\_3**:** **if** **(**m**>=**1**)** **if** **(!((**R**[**n**-**1**][**m**-**1**])|(**R**[**n**][**m**-**1**])|(**R**[**n**+**1**][**m**-**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

C\_4**:** **if** **(**m**>=**2**)** **if** **(!((**R**[**n**-**1**][**m**-**2**])|(**R**[**n**][**m**-**2**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

D\_1**:** **if** **(**m**>=**1**)** **if** **(!((**R**[**n**-**1**][**m**-**1**])|(**R**[**n**][**m**-**1**])|(**R**[**n**+**1**][**m**-**1**])|(**R**[**n**+**2**][**m**-**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

D\_2**:** **if** **(**m**>=**2**)** **if** **(!(**R**[**n**][**m**-**2**]))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

E\_1**:** **if** **(**m**>=**2**)** **if** **(!((**R**[**n**-**1**][**m**-**1**])|(**R**[**n**][**m**-**2**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

E\_2**:** **if** **(**m**>=**2**)** **if** **(!((**R**[**n**-**1**][**m**-**1**])|(**R**[**n**][**m**-**2**])|(**R**[**n**+**1**][**m**-**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

E\_3**:** **if** **(**m**>=**2**)** **if** **(!((**R**[**n**][**m**-**2**])|(**R**[**n**+**1**][**m**-**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

E\_4**:** **if** **(**m**>=**1**)** **if** **(!((**R**[**n**-**1**][**m**-**1**])|(**R**[**n**][**m**-**1**])|(**R**[**n**+**1**][**m**-**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

F\_1**:** **if** **(**m**>=**1**)** **if** **(!((**R**[**n**-**1**][**m**])|(**R**[**n**][**m**-**1**])|(**R**[**n**+**1**][**m**-**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

F\_2**:** **if** **(**m**>=**2**)** **if** **(!((**R**[**n**-**1**][**m**-**2**])|(**R**[**n**][**m**-**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

G\_1**:** **if** **(**m**>=**1**)** **if** **(!((**R**[**n**-**1**][**m**-**1**])|(**R**[**n**][**m**-**1**])|(**R**[**n**+**1**][**m**-**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

G\_2**:** **if** **(**m**>=**2**)** **if** **(!((**R**[**n**-**1**][**m**-**1**])|(**R**[**n**][**m**-**2**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

**default** MOVE\_ABLE**=**0**;**

**endcase**

**end**

**else** **if** **(**KEYBOARD**[**3**])** //RIGHT

**begin**

//MOVE\_ABLE=1;

**case** **(**BLOCK**)**

A\_1**:** **if** **(**m**<=**7**)** **if** **(!((**R**[**n**+**1**][**m**+**2**])|(**R**[**n**][**m**+**2**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

B\_1**:** **if** **(**m**<=**7**)** **if** **(!((**R**[**n**+**1**][**m**+**2**])|(**R**[**n**][**m**+**1**])|(**R**[**n**-**1**][**m**+**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

B\_2**:** **if** **(**m**<=**7**)** **if** **(!((**R**[**n**][**m**+**2**])|(**R**[**n**+**1**][**m**+**2**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**<=**0**;**

B\_3**:** **if** **(**m**<=**8**)** **if** **(!((**R**[**n**-**1**][**m**+**1**])|(**R**[**n**][**m**+**1**])|(**R**[**n**+**1**][**m**+**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

B\_4**:** **if** **(**m**<=**7**)** **if** **(!((**R**[**n**][**m**+**2**])|(**R**[**n**+**1**][**m**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

C\_1**:** **if** **(**m**<=**8**)** **if** **(!((**R**[**n**-**1**][**m**+**1**])|(**R**[**n**][**m**+**1**])|(**R**[**n**+**1**][**m**+**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

C\_2**:** **if** **(**m**<=**7**)** **if** **(!((**R**[**n**-**1**][**m**+**2**])|(**R**[**n**][**m**+**2**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

C\_3**:** **if** **(**m**<=**7**)** **if** **(!((**R**[**n**-**1**][**m**+**2**])|(**R**[**n**][**m**+**1**])|(**R**[**n**+**1**][**m**+**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

C\_4**:** **if** **(**m**<=**7**)** **if** **(!((**R**[**n**-**1**][**m**])|(**R**[**n**][**m**+**2**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

D\_1**:** **if** **(**m**<=**8**)** **if** **(!((**R**[**n**-**1**][**m**+**1**])|(**R**[**n**][**m**+**1**])|(**R**[**n**+**1**][**m**+**1**])|(**R**[**n**+**2**][**m**+**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

D\_2**:** **if** **(**m**<=**6**)** **if** **(!(**R**[**n**][**m**+**3**]))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

E\_1**:** **if** **(**m**<=**7**)** **if** **(!((**R**[**n**-**1**][**m**+**1**])|(**R**[**n**][**m**+**2**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

E\_2**:** **if** **(**m**<=**8**)** **if** **(!((**R**[**n**-**1**][**m**+**1**])|(**R**[**n**][**m**+**1**])|(**R**[**n**+**1**][**m**+**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

E\_3**:** **if** **(**m**<=**7**)** **if** **(!((**R**[**n**][**m**+**2**])|(**R**[**n**+**1**][**m**+**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

E\_4**:** **if** **(**m**<=**7**)** **if** **(!((**R**[**n**-**1**][**m**+**1**])|(**R**[**n**][**m**+**2**])|(**R**[**n**+**1**][**m**+**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

F\_1**:** **if** **(**m**<=**7**)** **if** **(!((**R**[**n**-**1**][**m**+**2**])|(**R**[**n**][**m**+**2**])|(**R**[**n**+**1**][**m**+**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

F\_2**:** **if** **(**m**<=**7**)** **if** **(!((**R**[**n**-**1**][**m**+**1**])|(**R**[**n**][**m**+**2**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

G\_1**:** **if** **(**m**<=**7**)** **if** **(!((**R**[**n**-**1**][**m**+**1**])|(**R**[**n**][**m**+**2**])|(**R**[**n**+**1**][**m**+**2**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

G\_2**:** **if** **(**m**<=**7**)** **if** **(!((**R**[**n**-**1**][**m**+**2**])|(**R**[**n**][**m**+**1**])))** MOVE\_ABLE**=**1**;** **else** MOVE\_ABLE**=**0**;**

**default** MOVE\_ABLE**=**0**;**

**endcase**

**end**

**end**

**else**

MOVE\_ABLE **=** 0**;**

**end**

// M\_OUT

**assign** M\_OUT **=** **{**R**[**23**],**R**[**22**],**R**[**21**],**R**[**20**],**R**[**19**],**R**[**18**],**R**[**17**],**R**[**16**],**R**[**15**],**R**[**14**],**R**[**13**],**R**[**12**],**R**[**11**],**R**[**10**],**R**[**9**],**R**[**8**],**R**[**7**],**R**[**6**],**R**[**5**],**R**[**4**],**R**[**3**],**R**[**2**],**R**[**1**],**R**[**0**]};**

// R

**integer** i**,**j**;**

**always** **@** **(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

**begin**

**for** **(**i **=** 0**;** i **<** 24**;** i **=** i **+** 1**)** R**[**i**]** **<=** 0**;**

REMOVE\_FINISH**<=**0**;**

**end**

**else** **if** **(**REMOVE\_1**)**

**begin**

**case** **(**BLOCK**)**

A\_1**:** **begin** R**[**n**][**m**]<=**1**;**R**[**n**][**m**+**1**]<=**1**;**R**[**n**+**1**][**m**]<=**1**;**R**[**n**+**1**][**m**+**1**]<=**1**;end**

B\_1**:** **begin** R**[**n**-**1**][**m**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**+**1**][**m**]<=**1**;**R**[**n**+**1**][**m**+**1**]<=**1**;end**

B\_2**:** **begin** R**[**n**-**1**][**m**+**1**]<=**1**;**R**[**n**][**m**-**1**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**][**m**+**1**]<=**1**;end**

B\_3**:** **begin** R**[**n**-**1**][**m**-**1**]<=**1**;**R**[**n**-**1**][**m**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**+**1**][**m**]<=**1**;end**

B\_4**:** **begin** R**[**n**][**m**-**1**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**][**m**+**1**]<=**1**;**R**[**n**+**1**][**m**-**1**]<=**1**;end**

C\_1**:** **begin** R**[**n**-**1**][**m**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**+**1**][**m**]<=**1**;**R**[**n**+**1**][**m**-**1**]<=**1**;end**

C\_2**:** **begin** R**[**n**][**m**-**1**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**][**m**+**1**]<=**1**;**R**[**n**+**1**][**m**+**1**]<=**1**;end**

C\_3**:** **begin** R**[**n**-**1**][**m**]<=**1**;**R**[**n**-**1**][**m**+**1**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**+**1**][**m**]<=**1**;end**

C\_4**:** **begin** R**[**n**-**1**][**m**-**1**]<=**1**;**R**[**n**][**m**-**1**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**][**m**+**1**]<=**1**;end**

D\_1**:** **begin** R**[**n**-**1**][**m**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**+**1**][**m**]<=**1**;**R**[**n**+**2**][**m**]<=**1**;end**

D\_2**:** **begin** R**[**n**][**m**-**1**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**][**m**+**1**]<=**1**;**R**[**n**][**m**+**2**]<=**1**;end**

E\_1**:** **begin** R**[**n**-**1**][**m**]<=**1**;**R**[**n**][**m**-**1**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**][**m**+**1**]<=**1**;end**

E\_2**:** **begin** R**[**n**-**1**][**m**]<=**1**;**R**[**n**][**m**-**1**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**+**1**][**m**]<=**1**;end**

E\_3**:** **begin** R**[**n**][**m**-**1**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**][**m**+**1**]<=**1**;**R**[**n**+**1**][**m**]<=**1**;end**

E\_4**:** **begin** R**[**n**-**1**][**m**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**][**m**+**1**]<=**1**;**R**[**n**+**1**][**m**]<=**1**;end**

F\_1**:** **begin** R**[**n**-**1**][**m**+**1**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**][**m**+**1**]<=**1**;**R**[**n**+**1**][**m**]<=**1**;end**

F\_2**:** **begin** R**[**n**-**1**][**m**-**1**]<=**1**;**R**[**n**-**1**][**m**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**][**m**+**1**]<=**1**;end**

G\_1**:** **begin** R**[**n**-**1**][**m**]<=**1**;**R**[**n**][**m**]<=**1**;**R**[**n**][**m**+**1**]<=**1**;**R**[**n**+**1**][**m**+**1**]<=**1**;end**

G\_2**:** **begin** R**[**n**-**1**][**m**]<=**1**;**R**[**n**-**1**][**m**+**1**]<=**1**;**R**[**n**][**m**-**1**]<=**1**;**R**[**n**][**m**]<=**1**;end**

**default**

**begin**

**for** **(**i **=** 0**;** i **<** 24**;** i **=** i **+** 1**)**

R**[**i**]** **<=** R**[**i**];**

**end**

**endcase**

REMOVE\_2\_S**<=**4'b1111**;**

**end**

**else** **if** **(**REMOVE\_2**)**

**begin**

**if** **(!**REMOVE\_FINISH**[**0**])**

**begin** **if** **((&**R**[**n**-**1**])|(**SIG**))**

**begin**

**if** **(**REMOVE\_2\_S**[**0**])** **begin** REMOVE\_2\_C**<=**n**-**1**;** REMOVE\_2\_S**[**0**]<=**0**;** SIG**<=**1**;end**

**else** **begin**

**if** **(**REMOVE\_2\_C**>=**1**)** **begin** R**[**REMOVE\_2\_C**]<=**R**[**REMOVE\_2\_C**-**1**];** REMOVE\_2\_C**<=**REMOVE\_2\_C**-**1**;** SIG**<=**1**;end**

**else** **begin** REMOVE\_FINISH**[**0**]<=**1**;**SIG**<=**0**;end**

**end**

**end**

**else** **begin** REMOVE\_FINISH**[**0**]<=**1**;** SIG**<=**0**;** **end**

**end**

**else** **if** **(!**REMOVE\_FINISH**[**1**])**

**begin** **if** **((&**R**[**n**])|(**SIG**))**

**begin**

**if** **(**REMOVE\_2\_S**[**1**])** **begin** REMOVE\_2\_C**<=**n**;** REMOVE\_2\_S**[**1**]<=**0**;** SIG**<=**1**;** **end**

**else** **begin**

**if** **(**REMOVE\_2\_C**>=**1**)** **begin** R**[**REMOVE\_2\_C**]<=**R**[**REMOVE\_2\_C**-**1**];** REMOVE\_2\_C**<=**REMOVE\_2\_C**-**1**;** SIG**<=**1**;** **end**

**else** **begin** REMOVE\_FINISH**[**1**]<=**1**;** SIG**<=**0**;** **end**

**end**

**end**

**else** **begin** REMOVE\_FINISH**[**1**]<=**1**;** SIG**<=**0**;** **end**

**end**

**else** **if** **(!**REMOVE\_FINISH**[**2**])**

**begin**

**if** **(**n**<=**22**)**

**begin** **if** **((&**R**[**n**+**1**])|(**SIG**))**

**begin**

**if** **(**REMOVE\_2\_S**[**2**])** **begin** REMOVE\_2\_C**<=**n**+**1**;** REMOVE\_2\_S**[**2**]<=**0**;**SIG**<=**1**;** **end**

**else** **begin**

**if** **(**REMOVE\_2\_C**>=**1**)** **begin** R**[**REMOVE\_2\_C**]<=**R**[**REMOVE\_2\_C**-**1**];** REMOVE\_2\_C**<=**REMOVE\_2\_C**-**1**;** SIG**<=**1**;** **end**

**else** **begin** REMOVE\_FINISH**[**2**]<=**1**;** SIG**<=**0**;** **end**

**end**

**end**

**else** **begin** REMOVE\_FINISH**[**2**]<=**1**;** SIG**<=**0**;** **end**

**end**

**else** **begin** REMOVE\_FINISH**[**2**]<=**1**;** SIG**<=**0**;** **end**

**end**

**else** **if** **(!**REMOVE\_FINISH**[**3**])**

**begin**

**if** **(**n**<=**21**)**

**begin** **if** **((&**R**[**n**+**2**])|(**SIG**))**

**begin**

**if** **(**REMOVE\_2\_S**[**3**])** **begin** REMOVE\_2\_C**<=**n**+**2**;** REMOVE\_2\_S**[**3**]<=**0**;** SIG**<=**1**;** **end**

**else** **begin**

**if** **(**REMOVE\_2\_C**>=**1**)** **begin** R**[**REMOVE\_2\_C**]<=**R**[**REMOVE\_2\_C**-**1**];** REMOVE\_2\_C**<=**REMOVE\_2\_C**-**1**;** SIG**<=**1**;** **end**

**else** **begin** REMOVE\_FINISH**[**3**]<=**1**;** SIG**<=**1**;** **end**

**end**

**end**

**else** **begin** REMOVE\_FINISH**[**3**]<=**1**;** SIG**<=**0**;** **end**

**end**

**else** **begin** REMOVE\_FINISH**[**3**]<=**1**;** SIG**<=**0**;** **end**

**end**

**else**

**begin**

**for** **(**i**=**0**;** i **<**24**;** i **=** i **+** 1**)** R**[**i**]** **<=** R**[**i**];**

REMOVE\_FINISH**<=**0**;**

SIG**<=**0**;**

**end**

**end**

**else** **if** **(**STOP**)** **for** **(**i**=**0**;**i**<=**23**;**i**=**i**+**1**)** R**[**i**]<=**0**;**

**end**

//BLOCK\_P

**always** **@** **(\*)**

**begin**

**case** **(**BLOCK**)**

A\_1**:** BLOCK\_P **=** A\_1**;**

B\_1**:** BLOCK\_P **=** B\_2**;**

B\_2**:** BLOCK\_P **=** B\_3**;**

B\_3**:** BLOCK\_P **=** B\_4**;**

B\_4**:** BLOCK\_P **=** B\_1**;**

C\_1**:** BLOCK\_P **=** C\_2**;**

C\_2**:** BLOCK\_P **=** C\_3**;**

C\_3**:** BLOCK\_P **=** C\_4**;**

C\_4**:** BLOCK\_P **=** C\_1**;**

D\_1**:** BLOCK\_P **=** D\_2**;**

D\_2**:** BLOCK\_P **=** D\_1**;**

E\_1**:** BLOCK\_P **=** E\_2**;**

E\_2**:** BLOCK\_P **=** E\_3**;**

E\_3**:** BLOCK\_P **=** E\_4**;**

E\_4**:** BLOCK\_P **=** E\_1**;**

F\_1**:** BLOCK\_P **=** F\_2**;**

F\_2**:** BLOCK\_P **=** F\_1**;**

G\_1**:** BLOCK\_P **=** G\_2**;**

G\_2**:** BLOCK\_P **=** G\_1**;**

**default** BLOCK\_P **=** 7'b0000000**;**

**endcase**

**end**

// BLOCK

**always** **@** **(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

BLOCK **<=** 7'b0000000**;**

**else** **if** **(**NEW**)**

BLOCK **<=** NEW\_BLOCK**;**

**else** **if** **(**SHIFT **&&** KEYBOARD**[**0**])**

BLOCK **<=** BLOCK\_P**;**

**else**

BLOCK **<=** BLOCK**;**

**end**

// DOWN\_ABLE

**always** **@** **(\*)**

**begin**

// if (!rst\_n)

DOWN\_ABLE **=** 0**;**

**if** **(**DOWN**)**

**begin**

//DOWN\_ABLE<=1;

**case** **(**BLOCK**)**

A\_1**:** **if** **(**n**<=**21**)** **begin** **if** **(!(**R**[**n**+**2**][**m**]** **|** R**[**n**+**2**][**m**+**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

B\_1**:** **if** **(**n**<=**21**)** **begin** **if** **(!(**R**[**n**+**2**][**m**]** **|** R**[**n**+**2**][**m**+**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

B\_2**:** **if** **(**n**<=**22**)** **begin** **if** **(!(**R**[**n**+**1**][**m**]** **|** R**[**n**+**1**][**m**-**1**]** **|** R**[**n**+**1**][**m**+**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

B\_3**:** **if** **(**n**<=**21**)** **begin** **if** **(!(**R**[**n**+**2**][**m**]** **|** R**[**n**][**m**-**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

B\_4**:** **if** **(**n**<=**21**)** **begin** **if** **(!(**R**[**n**+**1**][**m**]** **|** R**[**n**+**1**][**m**+**1**]** **|** R**[**n**+**2**][**m**-**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

C\_1**:** **if** **(**n**<=**21**)** **begin** **if** **(!(**R**[**n**+**2**][**m**]** **|** R**[**n**+**2**][**m**-**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

C\_2**:** **if** **(**n**<=**21**)** **begin** **if** **(!(**R**[**n**+**1**][**m**]** **|** R**[**n**+**1**][**m**-**1**]** **|** R**[**n**+**2**][**m**+**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

C\_3**:** **if** **(**n**<=**21**)** **begin** **if** **(!(**R**[**n**+**2**][**m**]** **|** R**[**n**][**m**+**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

C\_4**:** **if** **(**n**<=**22**)** **begin** **if** **(!(**R**[**n**+**1**][**m**]** **|** R**[**n**+**1**][**m**-**1**]** **|** R**[**n**+**1**][**m**+**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

D\_1**:** **if** **(**n**<=**20**)** **begin** **if** **(!(**R**[**n**+**3**][**m**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

D\_2**:** **if** **(**n**<=**22**)** **begin** **if** **(!(**R**[**n**+**1**][**m**]** **|** R**[**n**+**1**][**m**-**1**]** **|** R**[**n**+**1**][**m**+**1**]** **|** R**[**n**+**1**][**m**+**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

E\_1**:** **if** **(**n**<=**22**)** **begin** **if** **(!(**R**[**n**+**1**][**m**]** **|** R**[**n**+**1**][**m**-**1**]** **|** R**[**n**+**1**][**m**+**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

E\_2**:** **if** **(**n**<=**21**)** **begin** **if** **(!(**R**[**n**+**2**][**m**]** **|** R**[**n**+**1**][**m**-**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

E\_3**:** **if** **(**n**<=**21**)** **begin** **if** **(!(**R**[**n**+**2**][**m**]** **|** R**[**n**+**1**][**m**-**1**]** **|** R**[**n**+**1**][**m**+**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

E\_4**:** **if** **(**n**<=**21**)** **begin** **if** **(!(**R**[**n**+**2**][**m**]** **|** R**[**n**+**1**][**m**+**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

F\_1**:** **if** **(**n**<=**21**)** **begin** **if** **(!(**R**[**n**+**2**][**m**]** **|** R**[**n**+**1**][**m**+**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

F\_2**:** **if** **(**n**<=**22**)** **begin** **if** **(!(**R**[**n**+**1**][**m**]** **|** R**[**n**][**m**-**1**]** **|** R**[**n**+**1**][**m**+**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

G\_1**:** **if** **(**n**<=**21**)** **begin** **if** **(!(**R**[**n**+**1**][**m**]** **|** R**[**n**+**2**][**m**+**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

G\_2**:** **if** **(**n**<=**22**)** **begin** **if** **(!(**R**[**n**+**1**][**m**]** **|** R**[**n**+**1**][**m**-**1**]** **|** R**[**n**][**m**+**1**]))** DOWN\_ABLE **=** 1**;** **else** DOWN\_ABLE **=** 0**;** **end** **else** DOWN\_ABLE**=**0**;**

**default** DOWN\_ABLE **=** 0**;**

**endcase**

**end**

**else**

DOWN\_ABLE **=** 0**;**

**end**

// n

**always** **@** **(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

n **<=** 0**;**

**else** **if** **(**NEW**)**

n **<=** 1**;**

**else** **if** **((**SHIFT**)&(**AUTODOWN**))**

n**<=**n**+**1**;**

**else** **if** **((**SHIFT**)&(**KEYBOARD**[**1**]))**

n **<=** n **+** 1**;**

**else**

n **<=** n**;**

**end**

// m

**always** **@** **(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

m **<=** 0**;**

**else** **if** **(**NEW**)**

m **<=** 5**;**

**else** **if** **(**SHIFT**)**

**begin**

**if** **(**AUTODOWN**)** m**<=**m**;**

**else** **if** **(**KEYBOARD**[**2**])**

m **<=** m **-** 1**;**

**else** **if** **(**KEYBOARD**[**3**])**

m **<=** m **+** 1**;**

**else**

m **<=** m**;**

**end**

**else**

m **<=** m**;**

**end**

// NEW\_BLOCK

**always** **@(\*)**

**begin**

**if** **(!**rst\_n**)**

NEW\_BLOCK **=** A\_1**;**

**else** **if** **(**NEW**)**

**begin**

**case** **(**RAN**)**

1**:** NEW\_BLOCK **=** A\_1**;**

2**:** NEW\_BLOCK **=** B\_1**;**

3**:** NEW\_BLOCK **=** C\_1**;**

4**:** NEW\_BLOCK **=** D\_1**;**

5**:** NEW\_BLOCK **=** E\_1**;**

6**:** NEW\_BLOCK **=** F\_1**;**

7**:** NEW\_BLOCK **=** G\_1**;**

**default** NEW\_BLOCK **=** A\_1**;**

**endcase**

**end**

**else**

NEW\_BLOCK **=** A\_1**;**

**end**

// SHIFT\_FINISH

**always** **@(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

SHIFT\_FINISH **<=** 0**;**

**else** **if** **(**SHIFT**)**

SHIFT\_FINISH **<=** 1**;**

**else**

SHIFT\_FINISH **<=** 0**;**

**end**

// REMOVE\_2\_FINISH

**always** **@(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

REMOVE\_2\_FINISH **<=** 0**;**

**else** **if** **(&**REMOVE\_FINISH**)**

REMOVE\_2\_FINISH **<=** 1**;**

**else**

REMOVE\_2\_FINISH **<=** 0**;**

**end**

// remove\_cnt

**always** **@(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

remove\_cnt **<=** 0**;**

**else** **if** **((**remove\_cnt **<** 23**)** **&&** **(**REMOVE\_2 **==** 1**))**

remove\_cnt **<=** remove\_cnt **+** 1**;**

**else**

remove\_cnt **<=** 0**;**

**end**

// DIE\_TRUE

**always** **@(\*)**

**begin**

**if** **(**DIE**)** **begin**

**if** **(|**R**[**3**])** DIE\_TRUE **=** 1**;**

**else** DIE\_TRUE **=** 0**;**

**end**

**else** DIE\_TRUE**=**0**;**

**end**

**endmodule**

VGA模块

VGA\_top.v

**module** top**(**

clk**,**

rst**,**

number**,**

hsync\_r**,**

vsync\_r**,**

OutRed**,**

OutGreen**,**

OutBlue**);**

**output** hsync\_r**,** vsync\_r**;**

**output** **[**3**:**0**]**OutRed**,** OutGreen**;**

**output** **[**3**:**0**]**OutBlue**;**

**input** **[**199**:**0**]**number**;**

**input** clk**,**rst**;**

**wire** clk\_n**;**

clk\_unit myclk**(**

**.**clk**(**clk**),**

**.**rst**(**rst**),**

**.**clk\_n**(**clk\_n**)**

**);**

VGA myvga**(**

**.**hsync\_r**(**hsync\_r**),**

**.**vsync\_r**(**vsync\_r**),**

**.**OutRed**(**OutRed**),**

**.**OutGreen**(**OutGreen**),**

**.**OutBlue**(**OutBlue**),**

**.**clk\_n**(**clk\_n**),**

**.**rst**(**rst**),**

**.**num**(**number**)**

**);**

**endmodule**

clk\_unit.v

**module** clk\_unit**(**

clk**,**

rst**,**

clk\_n

**);**

**input** clk**,** rst**;**

**output** clk\_n**;**

**reg** clk\_n**;**

**reg** clk\_tmp**;**

**always** **@(posedge** clk\_tmp **or** **posedge** rst**)** **begin**

**if** **(**rst**)** **begin**

clk\_n **<=** 0**;**

**end**

**else** **begin**

clk\_n **<=** **~**clk\_n**;**

**end**

**end**

**always** **@(posedge** clk **or** **posedge** rst**)**

**begin**

**if** **(**rst**)**

clk\_tmp **<=** 0**;**

**else**

clk\_tmp **<=** **~**clk\_tmp**;**

**end**

**endmodule**

VGA.v

**module** VGA**(**

clk\_n**,**

rst**,**

hsync\_r**,**

vsync\_r**,**

OutRed**,**

OutGreen**,**

OutBlue**,**

num

**);**

**input** clk\_n**;**

**input** rst**;**

**input** **[**199**:**0**]** num**;**

**output** **reg** hsync\_r**;**

**output** **reg** vsync\_r**;**

**output[**3**:**0**]** OutRed**;**

**output[**3**:**0**]** OutGreen**;**

**output[**3**:**0**]** OutBlue**;**

**wire** **[**9**:**0**]** R **[**19**:**0**];**

**assign** R**[**0**]** **=** num**[**9**:**0**];**

**assign** R**[**1**]** **=** num**[**19**:**10**];**

**assign** R**[**2**]** **=** num**[**29**:**20**];**

**assign** R**[**3**]** **=** num**[**39**:**30**];**

**assign** R**[**4**]** **=** num**[**49**:**40**];**

**assign** R**[**5**]** **=** num**[**59**:**50**];**

**assign** R**[**6**]** **=** num**[**69**:**60**];**

**assign** R**[**7**]** **=** num**[**79**:**70**];**

**assign** R**[**8**]** **=** num**[**89**:**80**];**

**assign** R**[**9**]** **=** num**[**99**:**90**];**

**assign** R**[**10**]** **=** num**[**109**:**100**];**

**assign** R**[**11**]** **=** num**[**119**:**110**];**

**assign** R**[**12**]** **=** num**[**129**:**120**];**

**assign** R**[**13**]** **=** num**[**139**:**130**];**

**assign** R**[**14**]** **=** num**[**149**:**140**];**

**assign** R**[**15**]** **=** num**[**159**:**150**];**

**assign** R**[**16**]** **=** num**[**169**:**160**];**

**assign** R**[**17**]** **=** num**[**179**:**170**];**

**assign** R**[**18**]** **=** num**[**189**:**180**];**

**assign** R**[**19**]** **=** num**[**199**:**190**];**

**reg[**9**:**0**]**xsync**,**ysync**;**

**always** **@(posedge** clk\_n **or** **posedge** rst**)** **begin**

**if** **(**rst**)** **begin**

xsync **<=** 10'd0**;**

**end**

**else** **if** **(**xsync **==** 10'd799**)** **begin**

xsync **<=** 10'd0**;**

**end**

**else** **begin**

xsync **<=** xsync **+** 1**;**

**end**

**end**

**always** **@(posedge** clk\_n **or** **posedge** rst**)** **begin**

**if** **(**rst**)** **begin**

ysync **<=** 10'd0**;**

**end**

**else** **if** **(**ysync **==** 10'd524**)** **begin**

ysync **<=** 10'd0**;**

**end**

**else** **if** **(**xsync **==** 10'd799**)** **begin**

ysync **<=** ysync **+** 1**;**

**end**

**end**

**always** **@(posedge** clk\_n **or** **posedge** rst**)** **begin**

**if** **(**rst**)** **begin**

hsync\_r **<=** 1'b0**;**

**end**

**else** **if** **(**xsync **==** 799**)** **begin**

hsync\_r **<=**1'b0**;**

**end**

**else** **if** **(**xsync **==** 95**)** **begin**

hsync\_r **<=** 1'b1**;**

**end**

**end**

**always** **@(posedge** clk\_n **or** **posedge** rst**)** **begin**

**if** **(**rst**)** **begin**

vsync\_r **<=** 1'b0**;**

**end**

**else** **if** **(**ysync **==** 0**)** **begin**

vsync\_r **<=**1'b0**;**

**end**

**else** **if** **(**ysync **==** 1**)** **begin**

vsync\_r **<=** 1'b1**;**

**end**

**end**

**wire** valid**;**

**assign** valid **=** **(**xsync **>** 143**)** **&&** **(**xsync **<** 784**)** **&&** **(**ysync **>** 34**)** **&&** **(**ysync **<** 515**);**

**wire** **[**9**:**0**]**x\_pos**,** y\_pos**;**

**assign** x\_pos **=** xsync **-** 143**;**

**assign** y\_pos **=** ysync **-**34**;**

**wire** **[**9**:**0**]** x**;**

**wire** **[**19**:**0**]** y**;**

**assign** x**[**0**]** **=** **(**x\_pos **>=** 201**)** **&&** **(**x\_pos **<=** 224**);**

**assign** x**[**1**]** **=** **(**x\_pos **>=** 225**)** **&&** **(**x\_pos **<=** 248**);**

**assign** x**[**2**]** **=** **(**x\_pos **>=** 249**)** **&&** **(**x\_pos **<=** 272**);**

**assign** x**[**3**]** **=** **(**x\_pos **>=** 273**)** **&&** **(**x\_pos **<=** 296**);**

**assign** x**[**4**]** **=** **(**x\_pos **>=** 297**)** **&&** **(**x\_pos **<=** 320**);**

**assign** x**[**5**]** **=** **(**x\_pos **>=** 321**)** **&&** **(**x\_pos **<=** 344**);**

**assign** x**[**6**]** **=** **(**x\_pos **>=** 345**)** **&&** **(**x\_pos **<=** 368**);**

**assign** x**[**7**]** **=** **(**x\_pos **>=** 369**)** **&&** **(**x\_pos **<=** 392**);**

**assign** x**[**8**]** **=** **(**x\_pos **>=** 393**)** **&&** **(**x\_pos **<=** 416**);**

**assign** x**[**9**]** **=** **(**x\_pos **>=** 417**)** **&&** **(**x\_pos **<=** 440**);**

**assign** y**[**0**]** **=** **(**y\_pos **>=** 1**)** **&&** **(**y\_pos **<=** 24**);**

**assign** y**[**1**]** **=** **(**y\_pos **>=** 25**)** **&&** **(**y\_pos **<=** 48**);**

**assign** y**[**2**]** **=** **(**y\_pos **>=** 49**)** **&&** **(**y\_pos **<=** 72**);**

**assign** y**[**3**]** **=** **(**y\_pos **>=** 73**)** **&&** **(**y\_pos **<=** 96**);**

**assign** y**[**4**]** **=** **(**y\_pos **>=** 97**)** **&&** **(**y\_pos **<=**120**);**

**assign** y**[**5**]** **=** **(**y\_pos **>=** 121**)** **&&** **(**y\_pos **<=**144**);**

**assign** y**[**6**]** **=** **(**y\_pos **>=** 145**)** **&&** **(**y\_pos **<=**168**);**

**assign** y**[**7**]** **=** **(**y\_pos **>=** 169**)** **&&** **(**y\_pos **<=**192**);**

**assign** y**[**8**]** **=** **(**y\_pos **>=** 193**)** **&&** **(**y\_pos **<=**216**);**

**assign** y**[**9**]** **=** **(**y\_pos **>=** 217**)** **&&** **(**y\_pos **<=**240**);**

**assign** y**[**10**]** **=** **(**y\_pos **>=** 241**)** **&&** **(**y\_pos **<=**264**);**

**assign** y**[**11**]** **=** **(**y\_pos **>=** 265**)** **&&** **(**y\_pos **<=**288**);**

**assign** y**[**12**]** **=** **(**y\_pos **>=** 289**)** **&&** **(**y\_pos **<=**312**);**

**assign** y**[**13**]** **=** **(**y\_pos **>=** 313**)** **&&** **(**y\_pos **<=**336**);**

**assign** y**[**14**]** **=** **(**y\_pos **>=** 337**)** **&&** **(**y\_pos **<=**360**);**

**assign** y**[**15**]** **=** **(**y\_pos **>=** 361**)** **&&** **(**y\_pos **<=**384**);**

**assign** y**[**16**]** **=** **(**y\_pos **>=** 385**)** **&&** **(**y\_pos **<=**408**);**

**assign** y**[**17**]** **=** **(**y\_pos **>=** 409**)** **&&** **(**y\_pos **<=**432**);**

**assign** y**[**18**]** **=** **(**y\_pos **>=** 433**)** **&&** **(**y\_pos **<=**456**);**

**assign** y**[**19**]** **=** **(**y\_pos **>=** 457**)** **&&** **(**y\_pos **<=**480**);**

//wire [9:0]R [19:0];

//assign R[0] = 10'b0000000001;

//assign R[1] = 10'b0000000101;

//assign R[2] = 10'b0000010001;

//assign R[3] = 10'b0001000001;

//assign R[4] = 10'b0100000001;

//assign R[5] = 10'b0100000001;

//assign R[6] = 10'b0010000001;

//assign R[7] = 10'b0001000001;

//assign R[8] = 10'b0000100001;

//assign R[9] = 10'b0000010001;

//assign R[10] = 10'b0000001001;

//assign R[11] = 10'b0000000101;

//assign R[12] = 10'b0000000011;

//assign R[13] = 10'b0000000001;

//assign R[14] = 10'b0000001001;

//assign R[15] = 10'b0000100001;

//assign R[16] = 10'b0100000001;

//assign R[17] = 10'b0000000001;

//assign R[18] = 10'b1111111111;

//assign R[19] = 10'b1010101010;

**parameter** high **=** 12'b1111\_1111\_1111**;**

**reg** **[**11**:**0**]** vga\_rgb**;**

//integer j, k;

**always** **@(posedge** clk\_n **or** **posedge** rst**)** **begin**

**if** **(**rst**)** **begin**

vga\_rgb **<=** 0**;**

**end**

**else** **if** **(**valid**)**

**begin**

**if** **(**x\_pos**>=**201 **&&** x\_pos**<=**440**)**

**if** **(**x**[**0**]&**y**[**0**]&**R**[**0**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**0**]** **&**R**[**0**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**0**]** **&**R**[**0**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**0**]** **&**R**[**0**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**0**]** **&**R**[**0**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**0**]** **&**R**[**0**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**0**]** **&**R**[**0**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**0**]** **&**R**[**0**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**0**]** **&** R**[**0**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**0**]** **&** R**[**0**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]&**y**[**1**]&**R**[**1**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**1**]** **&**R**[**1**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**1**]** **&**R**[**1**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**1**]** **&**R**[**1**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**1**]** **&**R**[**1**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**1**]** **&**R**[**1**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**1**]** **&**R**[**1**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**1**]** **&**R**[**1**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**1**]** **&** R**[**1**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**1**]** **&** R**[**1**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]&**y**[**2**]&**R**[**2**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**2**]** **&**R**[**2**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**2**]** **&**R**[**2**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**2**]** **&**R**[**2**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**2**]** **&**R**[**2**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**2**]** **&**R**[**2**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**2**]** **&**R**[**2**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**2**]** **&**R**[**2**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**2**]** **&** R**[**2**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**2**]** **&** R**[**2**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]&**y**[**3**]&**R**[**3**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**3**]** **&**R**[**3**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**3**]** **&**R**[**3**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**3**]** **&**R**[**3**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**3**]** **&**R**[**3**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**3**]** **&**R**[**3**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**3**]** **&**R**[**3**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**3**]** **&**R**[**3**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**3**]** **&** R**[**3**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**3**]** **&** R**[**3**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]** **&** y**[**4**]** **&** R**[**4**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**4**]** **&**R**[**4**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**4**]** **&**R**[**4**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**4**]** **&**R**[**4**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**4**]** **&**R**[**4**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**4**]** **&**R**[**4**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**4**]** **&**R**[**4**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**4**]** **&**R**[**4**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**4**]** **&** R**[**4**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**4**]** **&** R**[**4**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]** **&**y**[**5**]** **&** R**[**5**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**5**]** **&** R**[**5**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**5**]** **&**R**[**5**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**5**]** **&**R**[**5**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**5**]** **&**R**[**5**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**5**]** **&**R**[**5**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**5**]** **&**R**[**5**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**5**]** **&**R**[**5**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**5**]** **&** R**[**5**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**5**]** **&** R**[**5**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]** **&** y**[**6**]** **&** R**[**6**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**6**]** **&** R**[**6**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**6**]** **&**R**[**6**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**6**]** **&**R**[**6**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**6**]** **&**R**[**6**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**6**]** **&**R**[**6**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**6**]** **&**R**[**6**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**6**]** **&**R**[**6**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**6**]** **&** R**[**6**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**6**]** **&** R**[**6**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]&**y**[**7**]&**R**[**7**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**7**]** **&**R**[**7**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**7**]** **&**R**[**7**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**7**]** **&**R**[**7**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**7**]** **&**R**[**7**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**7**]** **&**R**[**7**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**7**]** **&**R**[**7**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**7**]** **&**R**[**7**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**7**]** **&** R**[**7**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**7**]** **&** R**[**7**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]&**y**[**8**]&**R**[**8**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**8**]** **&**R**[**8**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**8**]** **&**R**[**8**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**8**]** **&**R**[**8**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**8**]** **&**R**[**8**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**8**]** **&**R**[**8**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**8**]** **&**R**[**8**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**8**]** **&**R**[**8**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**8**]** **&** R**[**8**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**8**]** **&** R**[**8**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]&**y**[**9**]&**R**[**9**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**9**]** **&**R**[**9**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**9**]** **&**R**[**9**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**9**]** **&**R**[**9**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**9**]** **&**R**[**9**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**9**]** **&**R**[**9**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**9**]** **&**R**[**9**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**9**]** **&**R**[**9**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**9**]** **&** R**[**9**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**9**]** **&** R**[**9**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]&**y**[**10**]&**R**[**10**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**10**]** **&**R**[**10**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**10**]** **&**R**[**10**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**10**]** **&**R**[**10**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**10**]** **&**R**[**10**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**10**]** **&**R**[**10**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**10**]** **&**R**[**10**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**10**]** **&**R**[**10**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**10**]** **&** R**[**10**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**10**]** **&** R**[**10**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]&**y**[**11**]&**R**[**11**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**11**]** **&**R**[**11**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**11**]** **&**R**[**11**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**11**]** **&**R**[**11**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**11**]** **&**R**[**11**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**11**]** **&**R**[**11**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**11**]** **&**R**[**11**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**11**]** **&**R**[**11**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**11**]** **&** R**[**11**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**11**]** **&** R**[**11**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]&**y**[**12**]&**R**[**12**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**12**]** **&**R**[**12**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**12**]** **&**R**[**12**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**12**]** **&**R**[**12**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**12**]** **&**R**[**12**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**12**]** **&**R**[**12**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**12**]** **&**R**[**12**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**12**]** **&**R**[**12**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**12**]** **&** R**[**12**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**12**]** **&** R**[**12**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]&**y**[**13**]&**R**[**13**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**13**]** **&**R**[**13**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**13**]** **&**R**[**13**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**13**]** **&**R**[**13**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**13**]** **&**R**[**13**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**13**]** **&**R**[**13**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**13**]** **&**R**[**13**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**13**]** **&**R**[**13**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**13**]** **&** R**[**13**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**13**]** **&** R**[**13**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]** **&** y**[**14**]** **&** R**[**14**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**14**]** **&**R**[**14**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**14**]** **&**R**[**14**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**14**]** **&**R**[**14**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**14**]** **&**R**[**14**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**14**]** **&**R**[**14**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**14**]** **&**R**[**14**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**14**]** **&**R**[**14**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**14**]** **&** R**[**14**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**14**]** **&** R**[**14**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]** **&**y**[**15**]** **&** R**[**15**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**15**]** **&** R**[**15**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**15**]** **&**R**[**15**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**15**]** **&**R**[**15**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**15**]** **&**R**[**15**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**15**]** **&**R**[**15**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**15**]** **&**R**[**15**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**15**]** **&**R**[**15**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**15**]** **&** R**[**15**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**15**]** **&** R**[**15**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]** **&** y**[**16**]** **&** R**[**16**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**16**]** **&** R**[**16**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**16**]** **&**R**[**16**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**16**]** **&**R**[**16**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**16**]** **&**R**[**16**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**16**]** **&**R**[**16**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**16**]** **&**R**[**16**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**16**]** **&**R**[**16**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**16**]** **&** R**[**16**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**16**]** **&** R**[**16**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]&**y**[**17**]&**R**[**17**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**17**]** **&**R**[**17**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**17**]** **&**R**[**17**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**17**]** **&**R**[**17**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**17**]** **&**R**[**17**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**17**]** **&**R**[**17**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**17**]** **&**R**[**17**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**17**]** **&**R**[**17**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**17**]** **&** R**[**17**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**17**]** **&** R**[**17**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]&**y**[**18**]&**R**[**18**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**18**]** **&**R**[**18**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**18**]** **&**R**[**18**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**18**]** **&**R**[**18**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**18**]** **&**R**[**18**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**18**]** **&**R**[**18**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**18**]** **&**R**[**18**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**18**]** **&**R**[**18**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**18**]** **&** R**[**18**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**18**]** **&** R**[**18**][**9**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**0**]&**y**[**19**]&**R**[**19**][**0**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**1**]** **&** y**[**19**]** **&**R**[**19**][**1**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**2**]** **&** y**[**19**]** **&**R**[**19**][**2**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**3**]** **&** y**[**19**]** **&**R**[**19**][**3**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**4**]** **&** y**[**19**]** **&**R**[**19**][**4**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**5**]** **&** y**[**19**]** **&**R**[**19**][**5**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**6**]** **&** y**[**19**]** **&**R**[**19**][**6**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**7**]** **&** y**[**19**]** **&**R**[**19**][**7**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**8**]** **&** y**[**19**]** **&** R**[**19**][**8**])**

vga\_rgb **<=** high**;**

**else** **if** **(**x**[**9**]** **&** y**[**19**]** **&** R**[**19**][**9**])**

vga\_rgb **<=** high**;**

**else**

vga\_rgb **<=** 12'b0000\_0000\_1111**;**

**else**

vga\_rgb **<=** 12'b0000\_0000\_0000**;**

**end**

**else**

**begin**

vga\_rgb **<=** 0**;**

**end**

**end**

**assign** OutRed **=** vga\_rgb**[**11**:**8**];**

**assign** OutGreen **=** vga\_rgb**[**7**:**4**];**

**assign** OutBlue **=** vga\_rgb**[**3**:**0**];**

**endmodule**

数据合并

merge.v

**module** merge **#(**

**parameter** ROW **=** 20**,**

**parameter** COL **=** 10

**)(**

**input** clk**,**

**input** rst\_n**,**

**input** **[(**ROW**+**4**)\***COL**-**1**:**0**]** data\_in**,**

**input** **[**6**:**0**]** shape**,**

**input** **[**3**:**0**]** x\_pos**,**

**input** **[**4**:**0**]** y\_pos**,**

**output** **[**ROW**\***COL**-**1**:**0**]** data\_out

**);**

// latency = 2 period

**wire** **[**8**:**0**]** loc**;**

**wire** **[**8**:**0**]** left**;**

**wire** **[**8**:**0**]** right**;**

**wire** **[**8**:**0**]** up**;**

**wire** **[**8**:**0**]** down**;**

**wire** **[**8**:**0**]** up\_left**;**

**wire** **[**8**:**0**]** up\_right**;**

**wire** **[**8**:**0**]** down\_left**;**

**wire** **[**8**:**0**]** down\_right**;**

**assign** loc **=** y\_pos**\***10**+**x\_pos**;**

**assign** left **=** loc **-** 1**;**

**assign** right **=** loc **+** 1**;**

**assign** up **=** loc **-** COL**;**

**assign** down **=** loc **+** COL**;**

**assign** up\_left **=** up **-** 1**;**

**assign** up\_right **=** up **+** 1**;**

**assign** down\_left **=** down **-** 1**;**

**assign** down\_right **=** down **+** 1**;**

**localparam** A1 **=** 7'b0001000**,**

B1 **=** 7'b0011000**,**

B2 **=** 7'b0010100**,**

B3 **=** 7'b0010010**,**

B4 **=** 7'b0010001**,**

C1 **=** 7'b0101000**,**

C2 **=** 7'b0100100**,**

C3 **=** 7'b0100010**,**

C4 **=** 7'b0100001**,**

D1 **=** 7'b0111000**,**

D2 **=** 7'b0110100**,**

E1 **=** 7'b1001000**,**

E2 **=** 7'b1000100**,**

E3 **=** 7'b1000010**,**

E4 **=** 7'b1000001**,**

F1 **=** 7'b1011000**,**

F2 **=** 7'b1010100**,**

G1 **=** 7'b1101000**,**

G2 **=** 7'b1100100**;**

**reg** **[**ROW**\***COL**-**1**:**0**]** data**;**

**reg** **[(**ROW**+**4**)\***COL**-**1**:**0**]** merge\_data**;**

**assign** data\_out **=** data **|** data\_in**[(**ROW**+**4**)\***COL**-**1**:**40**];**

//assign data\_out = data | data\_in[ROW\*COL-1:0];

**always** **@** **(posedge** clk **or** **negedge** rst\_n**)**

**begin**

**if** **(!**rst\_n**)**

data **<=** 0**;**

**else**

data **<=** merge\_data**[(**ROW**+**4**)\***COL**-**1**:**40**];**

// data <= merge\_data[ROW\*COL-1:0];

**end**

**always** **@** **(\*)**

**begin**

merge\_data **=** 0**;**

**case** **(**shape**)**

A1**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**right**]** **=** 1**;**

merge\_data**[**down**]** **=** 1**;**

merge\_data**[**down\_right**]** **=** 1**;**

**end**

B1**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**up**]** **=** 1**;**

merge\_data**[**down**]** **=** 1**;**

merge\_data**[**down\_right**]** **=** 1**;**

**end**

B2**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**left**]** **=** 1**;**

merge\_data**[**right**]** **=** 1**;**

merge\_data**[**up\_right**]** **=** 1**;**

**end**

B3**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**up**]** **=** 1**;**

merge\_data**[**up\_left**]** **=** 1**;**

merge\_data**[**down**]** **=** 1**;**

**end**

B4**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**left**]** **=** 1**;**

merge\_data**[**right**]** **=** 1**;**

merge\_data**[**down\_left**]** **=** 1**;**

**end**

C1**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**up**]** **=** 1**;**

merge\_data**[**down**]** **=** 1**;**

merge\_data**[**down\_left**]** **=** 1**;**

**end**

C2**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**left**]** **=** 1**;**

merge\_data**[**right**]** **=** 1**;**

merge\_data**[**down\_right**]** **=** 1**;**

**end**

C3**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**up**]** **=** 1**;**

merge\_data**[**down**]** **=** 1**;**

merge\_data**[**up\_right**]** **=** 1**;**

**end**

C4**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**left**]** **=** 1**;**

merge\_data**[**right**]** **=** 1**;**

merge\_data**[**up\_left**]** **=** 1**;**

**end**

D1**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**up**]** **=** 1**;**

merge\_data**[**down**]** **=** 1**;**

merge\_data**[**down**+**COL**]** **=** 1**;**

**end**

D2**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**left**]** **=** 1**;**

merge\_data**[**right**]** **=** 1**;**

merge\_data**[**right**+**1**]** **=** 1**;**

**end**

E1**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**left**]** **=** 1**;**

merge\_data**[**right**]** **=** 1**;**

merge\_data**[**up**]** **=** 1**;**

**end**

E2**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**left**]** **=** 1**;**

merge\_data**[**down**]** **=** 1**;**

merge\_data**[**up**]** **=** 1**;**

**end**

E3**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**down**]** **=** 1**;**

merge\_data**[**right**]** **=** 1**;**

merge\_data**[**left**]** **=** 1**;**

**end**

E4**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**up**]** **=** 1**;**

merge\_data**[**down**]** **=** 1**;**

merge\_data**[**right**]** **=** 1**;**

**end**

F1**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**up\_right**]** **=** 1**;**

merge\_data**[**down**]** **=** 1**;**

merge\_data**[**right**]** **=** 1**;**

**end**

F2**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**up**]** **=** 1**;**

merge\_data**[**up\_left**]** **=** 1**;**

merge\_data**[**right**]** **=** 1**;**

**end**

G1**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**up**]** **=** 1**;**

merge\_data**[**right**]** **=** 1**;**

merge\_data**[**down\_right**]** **=** 1**;**

**end**

G2**:**

**begin**

merge\_data**[**loc**]** **=** 1**;**

merge\_data**[**left**]** **=** 1**;**

merge\_data**[**up**]** **=** 1**;**

merge\_data**[**up\_right**]** **=** 1**;**

**end**

**default** merge\_data **=** 0**;**

**endcase**

**end**

**endmodule**

顶层模块

`timescale 1ns **/** 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 2015/11/15 19:38:52

// Design Name:

// Module Name: tetris

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

**module** tetris **#(**

**parameter** ROW **=** 20**,**

**parameter** COL **=** 10

**)(**

**input** clk**,**

**input** rst**,**

**input** UP\_KEY**,**

**input** LEFT\_KEY**,**

**input** RIGHT\_KEY**,**

**input** DOWN\_KEY**,**

**input** start**,**

**output** vsync\_r**,**

**output** hsync\_r**,**

**output** **[**3**:**0**]**OutRed**,** OutGreen**,**

**output** **[**3**:**0**]**OutBlue

**);**

**wire** **[**3**:**0**]** opcode**;**

**wire** gen\_random**;**

**wire** hold**;**

**wire** shift**;**

**wire** move\_down**;**

**wire** remove\_1**;**

**wire** remove\_2**;**

**wire** stop**;**

**wire** move**;**

**wire** isdie**;**

**wire** shift\_finish**;**

**wire** down\_comp**;**

**wire** move\_comp**;**

**wire** die**;**

**wire** **[**ROW**\***COL**-**1**:**0**]** data\_out**;**

**wire** **[**6**:**0**]** BLOCK**;**

**wire** **[**3**:**0**]** m**;**

**wire** **[**4**:**0**]** n**;**

**wire** **[(**ROW**+**4**)\***COL**-**1**:**0**]** M\_OUT**;**

**wire** rotate**;**

**wire** left**;**

**wire** right**;**

**wire** down**;**

**wire** auto\_down**;**

**wire** rst\_n**;**

**assign** rst\_n **=** **~**rst**;**

key u\_key **(**

**.**clk**(**clk**),**

**.**rst\_n**(**rst\_n**),**

**.**UP\_KEY**(**UP\_KEY**),**

**.**LEFT\_KEY**(**LEFT\_KEY**),**

**.**RIGHT\_KEY**(**RIGHT\_KEY**),**

**.**DOWN\_KEY**(**DOWN\_KEY**),**

**.**rotate**(**rotate**),**

**.**left**(**left**),**

**.**right**(**right**),**

**.**down**(**down**)**

**);**

game\_control\_unit u\_Controller **(**

**.**clk**(**clk**),**

**.**rst\_n**(**rst\_n**),**

**.**rotate**(**rotate**),**

**.**left**(**left**),**

**.**right**(**right**),**

**.**down**(**down**),**

**.**start**(**start**),**

**.**opcode**(**opcode**),**

**.**gen\_random**(**gen\_random**),**

**.**hold**(**hold**),**

**.**shift**(**shift**),**

**.**move\_down**(**move\_down**),**

**.**remove\_1**(**remove\_1**),**

**.**remove\_2**(**remove\_2**),**

**.**stop**(**stop**),**

**.**move**(**move**),**

**.**isdie**(**isdie**),**

**.**shift\_finish**(**shift\_finish**),**

**.**down\_comp**(**down\_comp**),**

**.**move\_comp**(**move\_comp**),**

**.**die**(**die**),**

**.**auto\_down**(**auto\_down**),**

**.**remove\_2\_finish**(**remove\_2\_finish**)**

**);**

Datapath\_Unit u\_Datapath **(**

**.**clk**(**clk**),**

**.**rst\_n**(**rst\_n**),**

**.**NEW**(**gen\_random**),**

**.**MOVE**(**move**),**

**.**DOWN**(**move\_down**),**

**.**DIE**(**isdie**),**

**.**SHIFT**(**shift**),**

**.**REMOVE\_1**(**remove\_1**),**

**.**REMOVE\_2**(**remove\_2**),**

**.**KEYBOARD**(**opcode**),**

**.**MOVE\_ABLE**(**move\_comp**),**

**.**SHIFT\_FINISH**(**shift\_finish**),**

**.**DOWN\_ABLE**(**down\_comp**),**

**.**DIE\_TRUE**(**die**),**

**.**M\_OUT**(**M\_OUT**),**

**.**n**(**n**),**

**.**m**(**m**),**

**.**BLOCK**(**BLOCK**),**

**.**REMOVE\_2\_FINISH**(**remove\_2\_finish**),**

**.**STOP**(**stop**),**

**.**AUTODOWN**(**auto\_down**)**

**);**

merge u\_merge **(**

**.**clk**(**clk**),**

**.**rst\_n**(**rst\_n**),**

**.**data\_in**(**M\_OUT**),**

**.**shape**(**BLOCK**),**

**.**x\_pos**(**m**),**

**.**y\_pos**(**n**),**

**.**data\_out**(**data\_out**)**

**);**

top u\_VGA **(**

**.**clk**(**clk**),**

**.**rst**(**rst**),**

**.**number**(**data\_out**),**

**.**hsync\_r**(**hsync\_r**),**

**.**vsync\_r**(**vsync\_r**),**

**.**OutRed**(**OutRed**),**

**.**OutGreen**(**OutGreen**),**

**.**OutBlue**(**OutBlue**)**

**);**

**endmodule**