SPRING
2022

CSEE4840 PROJECT

RIVER RAID III - DESIGN DOCUMENT

XINHAO SU, ROJAN BANMALI

1) Abstract

This design document describes the adaptation of Atari's classic River Raid game on an Altera DE1-SOC FPGA board. In our implementation, the graphics, audio and user inputs are handled by the FPGA hardware, whereas game logic, which includes, level design, player movement, enemy movement, collision detection and score keeping are done in software.

2) Design Overview

Figure 1 shows the organization of hardware modules and logic for the entire game. Figure 2 shows bit assignments for the memory layout used.

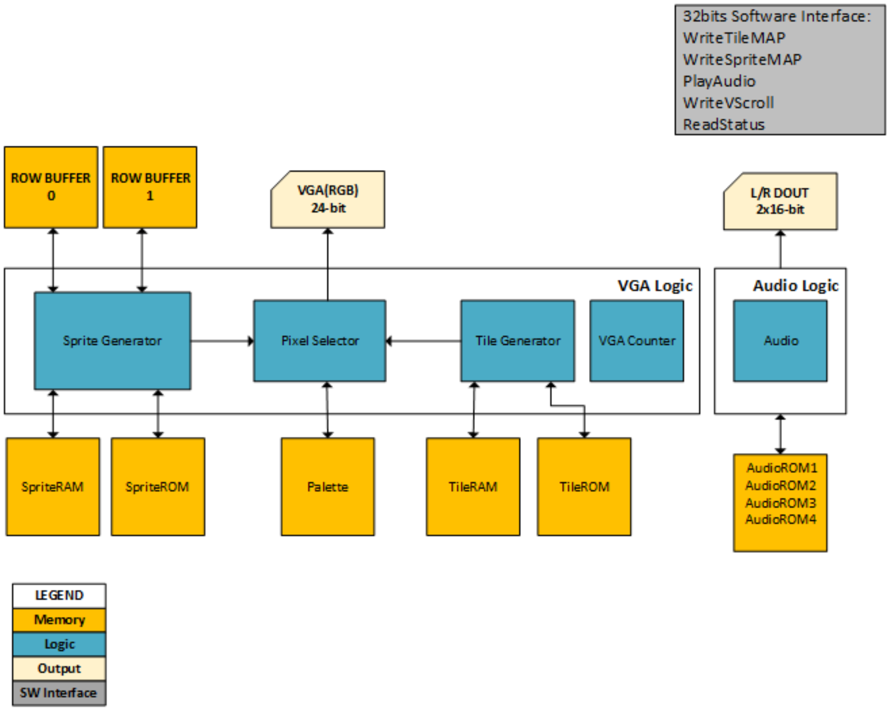


Figure 1 : Hardware components

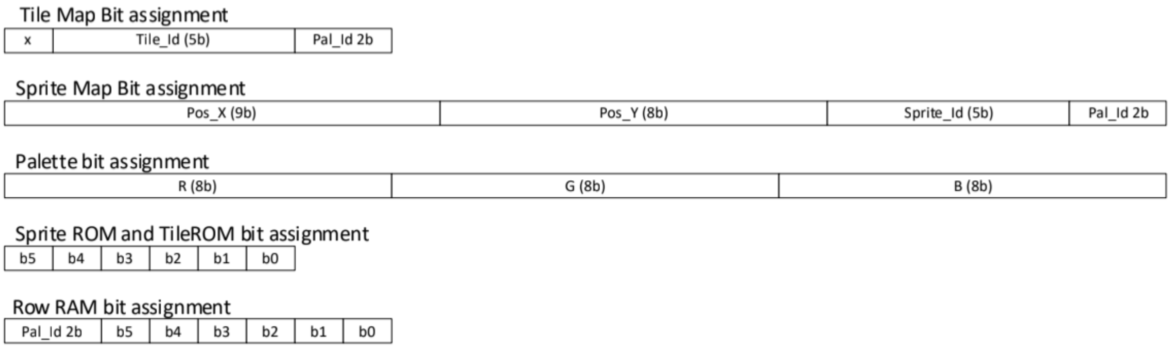


Figure 2 : Memory bit assignment

a. Memory Organization

The memory bit assignment used in the hardware is organized as follows:

i. Tile Map RAM
   1. Pal_Id (LSB): 2 bits Palette ID. This number selects palette from 4 available color palettes.
   2. Tile_Id: 5 bits Tile ID. This selects tile from 32 available 16x16 tile artworks.

ii. Sprite Map RAM
   1. Pal_id (LSB) 2 bits Palette ID. This number selects palette from 4 available color palettes.
   2. Sprite_Id: 5 bits Sprite ID. This selects tile from 32 available 16x16 sprite artworks.
   3. Pos_y: 8 bits Sprite Y coordinate.
   4. Pos_x: 9 bits Sprite X coordinate.

iii. Palette ROM
   1. Blue (LSB): 8 bits. Blue color value.
   2. Green (LSB): 8 bits. Green color value.
   3. Red (LSB): 8 bits. Red color value.

iv. Tile/ Sprite ROM
   1. px0_b5 to px0_b0: 6 bits color index. Next row defines pixel 1 value, next pixel 2, and so on. 6 bits will give 64 different shades.

v. Row Buffer:
   1. Color index: 6 bits (LSB)
   2. Pal_Id: 2 bits Palette ID.

b. Graphics Logic

Graphics Logic uses: VGA counter, Tile Map RAM, Tile ROM, Sprite Map RAM, Sprite ROM and Palette ROM as shown in Figure 1. The original resolution of the game is 160 x 192 pixels. However, our design uses 320 x 240 resolution, which is then stretched to fit 640 x 480 resolution by dividing horizontal and vertical counter by 2. The artwork of the game is stored as 6bit/pixel sprites and tiles in Sprite and Tile ROMs. The size of each sprite and tile is 16x16 pixels. The game has about 32 sprites and 32 tiles. Tiles are used to generate the scrolling background of the game, whereas sprites are used for the foreground objects of the game like player ship, enemy ship, missile, etc.

The following are the main components of the graphics logic:

i) VGA pixel selector:

VGA pixel selector selects pixel value from either sprite generator or tile generator and outputs the 24bit RGB value to the VGA DAC. If the sprite generator output has pixel value greater than 0, the VGA pixel selector selects the output of the sprite generator. It selects output of the tile generator otherwise.

ii) Vertical Scrolling Tiles generator:

The visible background of the game is arranged as 20 (columns) x 15 (rows) tiles, since the graphics resolution is 320 x 240 and the tile size is 16x16. To make vertical scrolling easier, there is an additional invisible row at the bottom of the screen making the background size 20 x 16 tiles as shown in Figure 3. The pseudo code for vertical scrolling is shown below Figure 3. In the figure, VScroll is an 8bit variable can be incremented or decremented to create scroll up or scroll down effect. As the vertical scrolling increments (scroll up) or decrements (scroll down), new tile map row data is sent by the software at the end of 16 vertical pixel boundary.

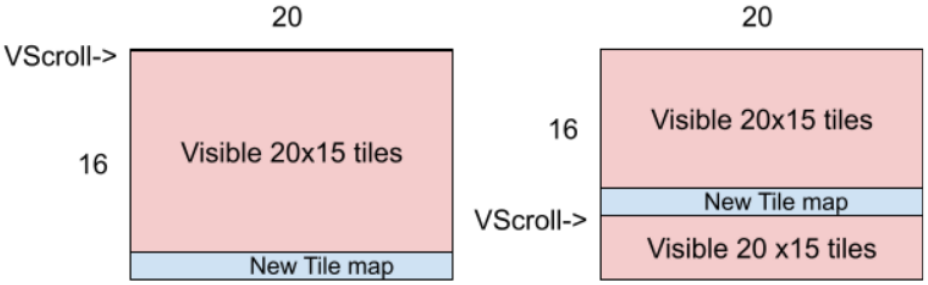


Figure 3 : Vertical Scrolling of background

```
TileMapRAM_address = hcount/16 + [(( vcount + VScroll)%256)/16]*20

TileMapData = TileMapRam [TileMapRAM_address]

TileROM_address=TileMapData[7:2]*256+(hcount%16)+
[(vcount+VScroll)%16*16]

TilePixelVal = TileROM [TileROM_address];
```

In the pseudo code above, Tile map RAM address is calculated based on hcount, vcount and VScroll. The Tile map RAM data is used to calculate

Tile ROM address. Tile ROM data is a 6 bit pixel value, which is mapped to one of four color pallets selected by 2bit palette id (Pal_id) defined in Tile map RAM in the VGA pixel selector to output a 24bit RGB color value.

iii) Sprite generator:

Sprite generator consists of four memory blocks: Sprite ROM, Sprite RAM, and two Row Buffers (RAMs): Row Buffer0 and Row Buffer1. Refer to 2.a.ii for sprite map bit field mapping. Sprite generator works by iterating through 16 Sprite RAM entries and comparing Y position of the sprite with next row to be drawn (vcount+1). If Y position of selected sprite equals vcount+1, the sprite generator starts writing (buffering) 16 horizontal pixel values of that sprite on one of the Row buffers starting at row buffer address defined by X position of the sprite. While the sprite generator is buffering pixels to write for the next row, it also outputs the pixels that had been previously buffered for the currently active row. For this, two row buffers, Row Buffer 0/1, are read from and written to in an alternating fashion. Furthermore, the iteration of Sprite RAM and buffering sprite pixels to Row RAM causes sprites defined in higher Sprite RAM location overlay on top of sprites defined in lower Sprite RAM locations.

c. Audio Logic

The game uses sample based audio logic. All audio samples used are 8bit 8 kHz mono samples of varying lengths. The following table shows four audio ROMs used and the trigger conditions for the samples. The audio DAC is configured by built in audio configuration block and is clocked by clock source with proper frequency generated by PLL block. The audio logic allows all sample to be played simultaneously.

	Sample (8hkz, mono)	Sample length	Tigger Condition
1	Fly	23213	Play sound in loop while flying
2	Fire	1599	Missile Fire
3	Refuel	2039	When going over flue tank
4	Explode	4095	When player or enemy is destroyed

Table 1: Audio Sample List

d. Player Input (Joystick) Logic

The game is designed to be played with an Atari Joystick connected to DE1-SOC GPIO_1 header. Figure 4 shows the pinout of Atari Joystick. Figure 5 shows pinout for DE1-SOC GPIO_1 header. Table 2 shows GPIO_1 connection used for the Joystick. The GPIOs have built in pull up resistors so external pull up resistors were not be necessary.

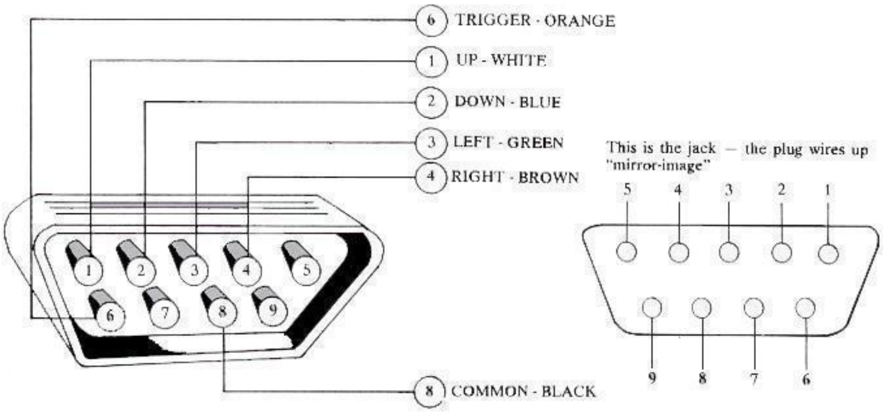


Figure 4 : Atari Joystick pinout

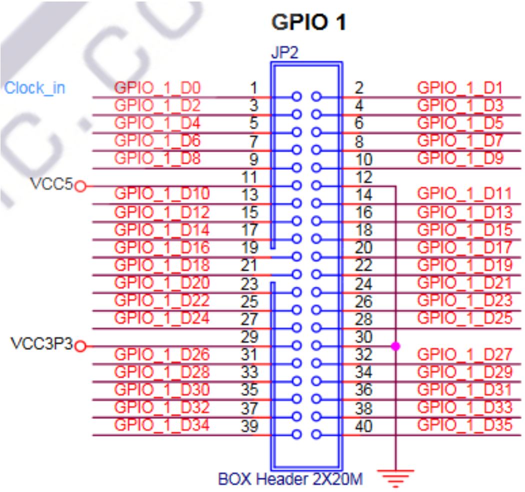


Figure 5 : DE1-SOC GPIO pins

GPIO 0 Pins	Pull up resistor	Atari Joystick Pins
30 (GND)		8
21 (GPIO_1_DB18)	internal	1 (UP)
22 (GPIO_1_DB19)	internal	2 (DOWN)
23 (GPIO_1_DB20)	internal	3 (LEFT)
24 (GPIO_1_DB21)	internal	4 (RIGHT)
25 (GPIO_1_DB22)	internal	6 (FIRE)

Table 2: GPIO_0 to Joystick pin assignments

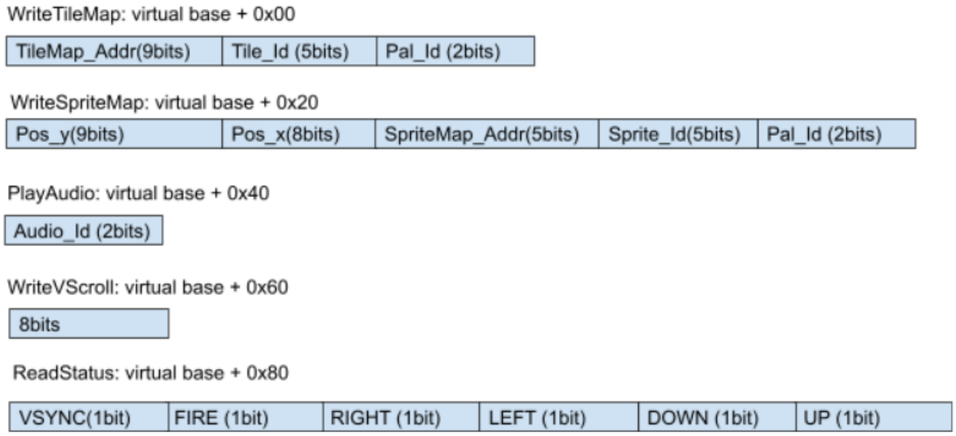
e. Software

The software for the game consists of linux driver defined in vga_ball.c, which handles the low level communication with the hardware, driver API defined in utils.c, which handles the communication between the main program and the driver, the game logic defined in rr.c, which defines the all game related functions, and the main game loop defined in main.c. Figure 6 shows high level view of the hardware (FPGA) and software interface.

i. Driver:

The data width of the data transfer to/from the software to/from the FPGA is 32 bits. Figure 6 shows commands and bit assignments of the data for each command available on the driver. The functions of the commands (interface) available on the driver are as follows:

- WriteTileMap command writes Tile_Id and Pal_Id (palette id) in Tile Map RAM at address TileMap_Addr.
- WriteSpriteMap command writes Pal_Id, Sprite_Id, Pos_x (x position) and Pos_y(y position) of the sprite in Sprite Map RAM at address SpriteMap_Addr.
- PlayAudio command plays audio sample assigned to Audio_Id.
- WriteVScroll command writes vertical scroll value.
- ReadStatus command reads the joystick switch status and update frame status. ReadStatus is called from the software in polling fashion.



WriteTileMap: virtual base + 0x00

TileMap_Addr(9bits)	Tile_Id (5bits)	Pal_Id (2bits)

WriteSpriteMap: virtual base + 0x20

Pos_y(9bits)	Pos_x(8bits)	SpriteMap_Addr(5bits)	Sprite_Id(5bits)	Pal_Id (2bits)

PlayAudio: virtual base + 0x40

Audio_Id (2bits)

WriteVScroll: virtual base + 0x60

8bits

ReadStatus: virtual base + 0x80

VSYNC(1bit)	FIRE (1bit)	RIGHT (1bit)	LEFT (1bit)	DOWN (1bit)	UP (1bit)

Figure 6 :Software Hardware interface commands

ii. **Game logic**:

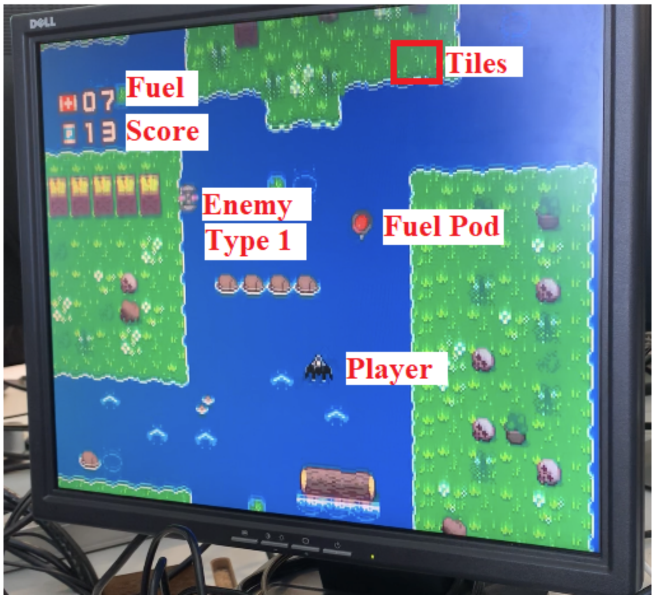


Figure 7 : Game Frame

- **Gameplay overview**:

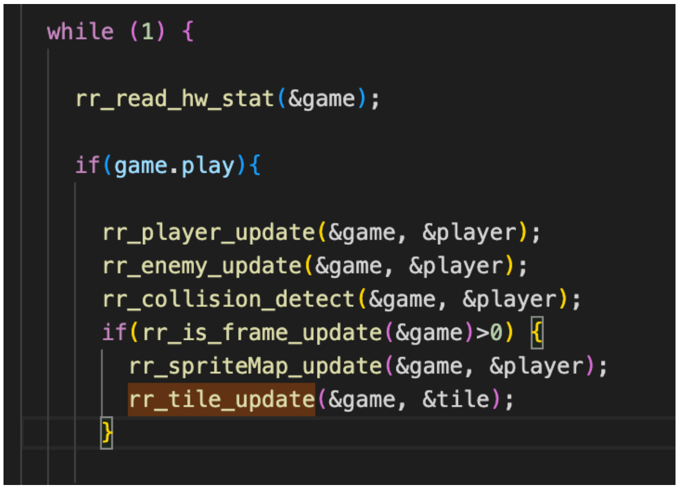
  The story line of the game is: The aliens have colonized the earth and cut down all of our forest. The player is on a mission to destroy all aliens on his stealth Tomahawk aircraft.

Figure 7 shows the main game screen. The game has ten full-screen background tile arrangements stored in a txt file. New tile data are read from the txt file and are written to Sprite RAM as the tiles scrolls down. When the file data pointer reaches the end, it is reset so the background screen repeats again.

In the game, the player has to destroy randomly generated moving enemy ships, by shooting missiles at them, and avoid colliding with them. The player can fire up to two missiles at a time. The player earns a point for every missile strike on an enemy. The player gets only one life, so if an enemy collides with the player; the game restarts from the beginning. In addition to avoiding collision, the player also has to maintain the fuel level by picking up moving fuel sprites generated at random time. The fuel level and game score are shown on top left of the screen. To make the game more interesting, enemies have attack mode which can activate randomly. In attack mode, the enemies track and try to crash themselves on to the player's ship. Furthermore, to make the game more interesting and challenging, more enemies are generated as the game progresses.

- **Game loop:**

  The main game loop runs in a dedicated thread. Code snippet, CS1, below shows the main game loop.



```
while (1) {

  rr_read_hw_stat(&game);

  if(game.play){

    rr_player_update(&game, &player);
    rr_enemy_update(&game, &player);
    rr_collision_detect(&game, &player);
    if(rr_is_frame_update(&game)>0) {
      rr_spriteMap_update(&game, &player);
      rr_tile_update(&game, &tile);
    }
```

CS 1 : Main Game loop (main.c)

In the game loop code -
rr_read_hw_stat(): Reads joystick and update_frame flag from the hardware.
rr_player_update(): Handles Joystick input and updates player's position and fuel level.
rr_enemy_update(): Creates new enemy ship (total of 5 enemies can exist at a time). Updates enemy position. Sets enemy attack mode.
rr_collision_detect(): Checks collision between enemy sprites and missiles, and enemy sprites and player sprite. Collision detection logic does a rectangular boundary check to see if boundaries of 16x16 sprites overlap.
rr_spriteMap_update(): Writes all updated sprite values to Sprite RAM via sprite RAM registers.
rr_tile_update(): Increments Vscroll register and writes Tile MAP RAM via tile map RAM register.
rr_is_frame_update() function makes sure that sprite map and tile map are updated during vertical blanking.

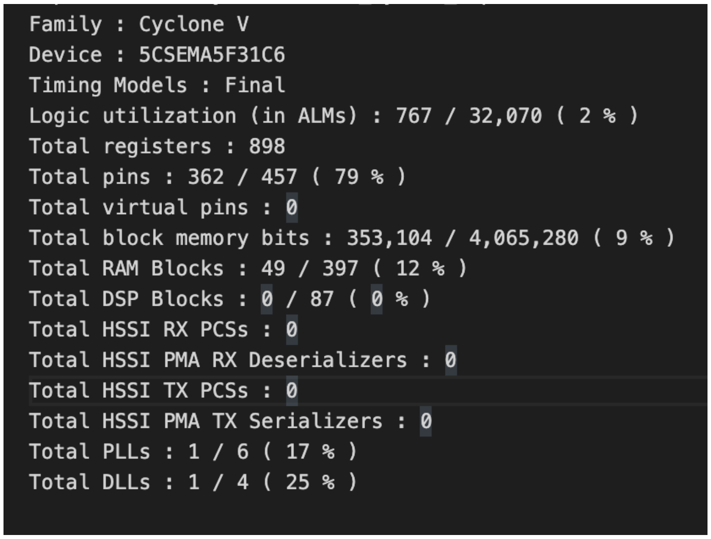
3) Memory Budget

Table 3 below shows memory budget estimated for the game.

Category	Per Size	Quantity	Size
Background Tile Map	8 bit	320 (20 row * 16 col) tiles	2,560 bits
Background Tile	16 * 16 * 5 (5bit color)	32 tile artwork	40,960 bits
Sprite Map	24 bit	32 sprites	768 bits
Sprite Assets	16 * 16 * 5 (5bit color)	32 sprite artwork	40,960 bits
Audio Sample	8 bit * 8KHz * 1 second	4 samples	256,000 bits
Palette	24bits (RGB) * 32 shades	4 palettes	3072 bits
Row RAM	8bits*512	2 Row buffers	8192 bits
Total (estimated)			352,512

Table 3: Memory Budget

Figure 8 shows the actual resources used by the final hardware design on the FPGA. Our design only used 9% of the total block memory bits, 12% of RAM blocks and 2% of the logic blocks.



```
Family : Cyclone V
Device : 5CSEMA5F31C6
Timing Models : Final
Logic utilization (in ALMs) : 767 / 32,070 ( 2 % )
Total registers : 898
Total pins : 362 / 457 ( 79 % )
Total virtual pins : 0
Total block memory bits : 353,104 / 4,065,280 ( 9 % )
Total RAM Blocks : 49 / 397 ( 12 % )
Total DSP Blocks : 0 / 87 ( 0 % )
Total HSSI RX PCSs : 0
Total HSSI PMA RX Deserializers : 0
Total HSSI TX PCSs : 0
Total HSSI PMA TX Serializers : 0
Total PLLs : 1 / 6 ( 17 % )
Total DLLs : 1 / 4 ( 25 % )
```

Figure 8 : Total resources used for final hardware

4) Improvements

From the memory usage, it is clear that we can easily improve the graphics of the game by increasing the resolution of the VGA output as well as the pixels of the sprites and tiles. Further, we can incorporate on chip SD RAM to store longer, 16-bit stereo high-resolution sound samples to improve the gaming experience. We could also make the artwork for sprites and tiles loadable from the software during the initialization of the game instead of using memory-initialization mif files, which permanently embeds the artwork on the hardware. Feature to scroll the game horizontally could also be added easily on the current design. This way, other sprite and tile based games can be implemented just from the software on the same hardware design.

5) Challenges and Lessons Learned

- Test bench - One thing we learnt from doing the project is that there is no escaping test bench. Learning how to properly setup and debug Verilog code using test bench early on is a must and will save a lot of time.
- Graphics update timing and frame rate - It took us a while to properly implement graphics update routine to get smooth animation without any screen or sprites flickering. We were able to achieve this by polling an update flag, which is asserted during VGA VSYNC period, and only updating the sprites and tiles during that period.

6) Contribution

Both of us (Rojan and Xinhao) contributed equally on this project. List below shows how we divided the work.

- Hardware
    - Tile Generator and Palette – Rojan, Xinhao
    - Sprite Generator –Rojan, Xinhao
    - Sprite and Tile artwork – Xinhao
    - Audio – Rojan
- Software
    - Driver – Rojan
    - Game functions – Rojan, Xinhao
    - Score and Fuel - Xinhao

7) References:

1) https://fileadmin.cs.lth.se/cs/Education/EDA385/HT11/student_doc/final_reports/spaceshooter.pdf
2) https://www.atariarchives.org/creativeatari/Joytricks.php
3) https://people.ece.cornell.edu/land/courses/ece5760/DE1_SOC/DE1-SoC%20schematic.pdf

8) Appendix A - Code:

vga_ball.h

```
/* module:vga_ball.h
 * Team - River Raid
 * Rojan Banmali (rb3199)
 * Xinhao Su (xs2413)
 *
 * CSEE4840
 * Stephen A. Edwards
 * Columbia University
 */


#ifndef _VGA_BALL_H
#define _VGA_BALL_H

#include <linux/ioctl.h>


typedef union {

  struct{
		uint32_t pallD:2;
    uint32_t tileID:5;
    uint32_t tileMAP_addr:9;
    uint32_t unused:16;
  }bits;
 uint32_t raw32;


} tileMAP_t;


typedef union {
 struct{
    uint32_t palID:2;
    uint32_t spriteID:5;
    uint32_t spriteMAP_addr:5;
            uint32_t x:9;
    uint32_t y:8;
    uint32_t unused:3;
 }bits;
 uint32_t raw32;
} spriteMAP_t;


typedef union {
 struct{
            uint32_t audioID:2;
    uint32_t unused:29;
 }bits;
 uint32_t raw32;


} audio_t;


typedef union {
 struct{
            uint32_t vscroll:8;
    uint32_t unused:24;
```

```
  }bits;
  uint32_t raw32;

} vscroll_t;

typedef union {
  struct{
            uint32_t joystick_UP:1;
    uint32_t joystick_DOWN:1;
    uint32_t joystick_LEFT:1;
    uint32_t joystick_RIGHT:1;
    uint32_t joystick_FIRE:1;
    uint32_t update_frame:1;
  }bits;
  uint32_t raw32;

} status_t;



typedef struct {

  tileMAP_t    tileMAP;
  spriteMAP_t  spriteMAP;
  audio_t      audio;
  vscroll_t    vscroll;
  status_t     status;

} vga_ball_arg_t;
```

```
#define VGA_BALL_MAGIC 'q'

/* ioctls and their arguments */
#define VGA_BALL_WRITE_TILE_MAP       _IOW(VGA_BALL_MAGIC, 1, vga_ball_arg_t *)
#define VGA_BALL_WRITE_SPRITE_MAP     _IOW(VGA_BALL_MAGIC, 2, vga_ball_arg_t *)
#define VGA_BALL_WRITE_AUDIO          _IOW(VGA_BALL_MAGIC, 3, vga_ball_arg_t *)
#define VGA_BALL_WRITE_VSCROLL        _IOW(VGA_BALL_MAGIC, 4, vga_ball_arg_t *)

#define VGA_BALL_READ_STATUS          _IOR(VGA_BALL_MAGIC, 5, vga_ball_arg_t *)

#endif
```

rr.c

```
/* module:rr.c
 * Team - River Raid
 * Rojan Banmali (rb3199)
 * Xinhao Su (xs2413)
 *
 * CSEE4840
 * Stephen A. Edwards
 * Columbia University
 */

#include "rr.h"
```

```
//TODO: make this part of the enemy and missile object
static rr_missile_t missile[PLAYER_MAX_MISSILE];
static rr_enemy_t enemy_table[GAME_MAX_ENEMY];

static void fuel_sprite_update(rr_game_t *gm, rr_player_t *pl);
static void score_sprite_update(rr_game_t *gm);
//static void fuel_update(rr_player_t *pl);
static void player_stop_game (rr_game_t *gm, rr_player_t *pl);



/*
* rr_tile_init()
*
*
*
*
*
*
*/
void rr_tile_init(rr_game_t*gm, rr_tile_t* tl){

  tl->line = 14;
  tl->addr = 20*tl->line;
  tl->vs.bits.vscroll = 0;
  tl->bound=0;
  set_vscroll(gm, &(tl->vs));
```

```
}

/*
*rr_tile_update
*
*
*
*
*/
void rr_tile_update(rr_game_t*gm, rr_tile_t* tl){

  int fdata;
  uint8_t tileid;
   gm->ticks++; //game timer

   tl->vs.bits.vscroll-=gm->vspd;

    if(tl->bound>15) {//tl->vs.bits.vscroll%16==0 //use bound variable instead of %
      tl->bound=0;
      tl->tm.raw32=0;
      for(int i=0;i<20;i++){

       if(fscanf(gm->tile_fp, "%d", &fdata)==EOF) {
        fseek(gm->tile_fp, 0, SEEK_SET);
        fscanf(gm->tile_fp, "%d", &fdata);
        // break;
       }
       //printf("fdata: %d\n",fdata); //DEBUG
       tileid = (uint8_t)fdata;
```

```
        tl->tm.raw32 = tl->addr<<7 | tileid;
        set_tileMAP(gm, &(tl->tm));

        tl->addr++;
      }
      //tileid = rand()%32;
      tl->line--;
      if(tl->line<0) tl->line = 15;
      tl->addr=tl->line*20;

    }
     tl->bound++;
    set_vscroll(gm, &(tl->vs));


}




/*
* rr_reset()
* Initialize player object
*
*
*
*
*
*/
```

```
void rr_reset (rr_game_t *gm, rr_player_t* pl){


  //reset tilemap file;
  fseek(gm->tile_fp, 0, SEEK_SET);
  gm->ticks=0;
  gm->score=0;
  gm->vspd=1;
  gm->reset_dash=1;


  srand(time(NULL)); //seed randgom gen.


  //init player sprite


  pl->ship.posX=(SCREEN_MAX_X/2-8);
  pl->ship.posY=SCREEN_MAX_Y-20;
  pl->ship.vx=0;
  pl->ship.vy=0;
  pl->fire_pressed=0;
  pl->ship.sp_num=PLAYER_SP_NUM;
  pl->ship.sp_id=PLAYER_SP_ID;
  pl->ship.sp_pid=PLAYER_SP_PID;
  pl->fuel=PLAYER_MAX_FUEL;
  pl->fuel_ticks=PLAYER_FUEL_TICKS;


  //init. sprite RAM
  for(int i=0;i<31;i++){
   set_spriteMAP(gm, i,0,0, GAME_MIN_X, GAME_MIN_Y);
  }
  //reset enemy
  for(int i=0; i<GAME_MAX_ENEMY; i++){
     enemy_table[i].ship.alive=0;
     enemy_table[i].ship.die_ticks=0;
   }

  //TODO: not possible to do something like pl->missile[i] so using local missile array.
  //TODO:remove this explicit use of missile array
  for(int i=0; i<PLAYER_MAX_MISSILE;i++){
   missile[i].ship.alive=0;
   missile[i].ship.die_ticks=0;
   missile[i].ship.posX=GAME_MIN_X;
   missile[i].ship.posY=GAME_MIN_Y;
  }

   set_spriteMAP(gm, pl->ship.sp_num,
              pl->ship.sp_id,
              pl->ship.sp_pid,
              pl->ship.posX,
              pl->ship.posY);

 set_audio(gm, AUDIO_FLYING);
}



/*
* rr_player_update
* Reads GPIO for Joystick
*
```

```
*

*

*

*

*/

static int flashing_tick = 0;

void rr_player_update(rr_game_t*gm, rr_player_t *pl)


{


  //status_t status;


  //Consume fuel

  pl->fuel_ticks--;

  if(pl->fuel_ticks<=0){


    #ifdef DEBUG_PRINT

     printf("Fuel--\n");

    #endif


     pl->fuel_ticks=PLAYER_FUEL_TICKS;

     pl->fuel--;




  }


  if (pl->fuel <= 5) {

       flashing_tick++;

        if (flashing_tick < 50) pl->ship.sp_pid =  PLAYER_SP_EMTPY_PID; //change color if low on fuel
```

```
      else pl->ship.sp_pid =  PLAYER_SP_PID; //change color if low on fuel
      flashing_tick = flashing_tick >= 100 ? 0 : flashing_tick;
   }
   if (pl->fuel > 5) pl->ship.sp_pid = PLAYER_SP_PID; //change color if low on fuel

   if(pl->fuel<=0){

     #ifdef DEBUG_PRINT
      printf("Emtpy Fuel\n");//DEBUG
     #endif

    player_stop_game(gm,pl);

   }
//check update missile
//TODO: not possible to do something like pl->missile[i] so using local missile array.

for(int i=0; i<PLAYER_MAX_MISSILE;i++){

   if(missile[i].ship.alive==0){
     continue;
   }

    missile[i].ship.posY+=missile[i].ship.vy;

    if (missile[i].ship.posY<=GAME_MIN_Y){
       missile[i].ship.alive=0;
      //DEBUG:printf("missile terminated\n");
     }
```

```
}

//read_status(gm, &status); //GPIO status

uint8_t st = 0x1F & (uint8_t)gm->hw_stat;//status.raw32;

pl->ship.vy= !(st & JOYSTICK_DOWN) * (PLAYER_SPEED_Y) +  !(st & JOYSTICK_UP) * (-PLAYER_SPEED_Y);
pl->ship.vx= !(st & JOYSTICK_RIGHT) * (PLAYER_SPEED_X) + !(st & JOYSTICK_LEFT) * (-PLAYER_SPEED_X);

pl->ship.posX = pl->ship.posX + pl->ship.vx;
pl->ship.posY = pl->ship.posY + pl->ship.vy;

if(pl->ship.posX > PLAYER_MAX_X) //clamp player x position
  pl->ship.posX=PLAYER_MAX_X;
else if (pl->ship.posX < PLAYER_MIN_X)
  pl->ship.posX=PLAYER_MIN_X;

if(pl->ship.posY > PLAYER_MAX_Y) //clamp player y position
  pl->ship.posY=PLAYER_MAX_Y;
else if (pl->ship.posY < PLAYER_MIN_Y)
  pl->ship.posY=PLAYER_MIN_Y;

//Check fire button and fire missile
if(!(st & JOYSTICK_FIRE) && !pl->fire_pressed) {
  //DEBUG: printf("Fire pressed\n");

  pl->fire_pressed=1;
```

```
   for(int i=0; i<PLAYER_MAX_MISSILE;i++){
    if(missile[i].ship.alive==1) continue;
    missile[i].ship.alive = 1;
    missile[i].ship.die_ticks=0;
    missile[i].ship.posX=pl->ship.posX;
    missile[i].ship.posY=pl->ship.posY;
    missile[i].ship.vy=-MISSILE_VY;
    missile[i].ship.vx=0;
    missile[i].ship.sp_id = MISSILE_SP_ID;
    missile[i].ship.sp_num = MISSILE_SP_NUM-i;
    missile[i].ship.sp_pid = MISSILE_SP_PID;

    set_audio(gm, AUDIO_FIRE);

   /*DEBUG: printf("missile fired:%d, x:%d, y:%d\n",
    missile[i].ship.sp_num,
    missile[i].ship.posX,
    missile[i].ship.posY);*/

    break;
   }

  }else if((st & JOYSTICK_FIRE)){
   pl->fire_pressed=0;

  }

  return;
```

```
}


/*
* rr_enemy_create
* Create Enemy
*
*
*
*
*
*/
void rr_enemy_update(rr_game_t*gm, rr_player_t*pl ){



 uint16_t x,y;
 int vx,vy;
 uint8_t max_enemy=1;

 gm->add_enemy_ticks++;

 if(gm->add_enemy_ticks>100){ // add enemy every 100 ticks

  gm->add_enemy_ticks=0;

 //make things more interesting as ticks++
  if(gm->ticks>6000){
     max_enemy = 6;
     gm->vspd = 2; //increase game speed
  }
  else if (gm->ticks>3000)
   max_enemy = 5;
  else if (gm->ticks>2000)
  {
   max_enemy = 3;
  }
  else if (gm->ticks>1000)
  max_enemy = 2;



  //create enemy

 for(int i=0; i<max_enemy;i++){

   if(enemy_table[i].ship.alive==1) continue;

    uint8_t r = (rand()%ENEMY_SP_TOTAL_NUM) + ENEMY_SP_ID_START; //ENEMY sprite_id 1-4
    uint8_t sp_id;
    if (r==ENEMY_SP_ID_START+ENEMY_SP_TOTAL_NUM-1) sp_id=FUEL_SP_ID; else sp_id=r; //make last enemy fuel

    uint16_t rand_posxy = rand()%1024;

    if(rand_posxy & 0x01){ // position horizontally

    x = rand_posxy>>1;

    if(x & 0x01){
      y = GAME_MAX_Y; //come from top
      vy=-ENEMY_SP_SPEED;
    }


     else{
      y = GAME_MIN_Y; //come from bottom
      vy=ENEMY_SP_SPEED;
     }


     if(x>GAME_MAX_X){
      x=GAME_MAX_X;
      vx=-ENEMY_SP_SPEED;
     }else{
      vx=1;
     }


    }else{ //position vertically


     y = rand_posxy>>1;


     if(y & 0x01){
      x = GAME_MAX_X; //come from RIGHT
      vx = -ENEMY_SP_SPEED;
     }
     else{
      x = GAME_MIN_X; //come from LEFT
      vx=ENEMY_SP_SPEED;
     }
```

```
        if(y>GAME_MAX_Y){
          x=GAME_MAX_Y;
          vy=-ENEMY_SP_SPEED;
        } else{
          vy=ENEMY_SP_SPEED;
        }
      }
      //starting parm. for new enemy
      enemy_table[i].ship.posX=x;
      enemy_table[i].ship.posY=y;
      enemy_table[i].ship.vx=vx;
      enemy_table[i].ship.vy=vy;
      enemy_table[i].ship.sp_id = sp_id;
      enemy_table[i].ship.sp_pid = ENEMY_SP_PID;
      enemy_table[i].ship.alive=1;
      enemy_table[i].ship.die_ticks=0;
      enemy_table[i].attack=0;
      enemy_table[i].attack_ticks=ENEMY_ATTACK_TICKS;

 }
 }

  int dx;
  int dy;
 //update enemy
 //TODO: use various motion profile for enemies
 for(int i=0; i<GAME_MAX_ENEMY; i++){

   if(enemy_table[i].ship.alive==0 ||
    (enemy_table[i].ship.alive==1 && enemy_table[i].ship.die_ticks>0)){
     continue;
   }

   //Lame attack code
   //TODO: enemies clutter up near the player. improve. maybe add motion profile.

   enemy_table[i].attack_ticks--;

   if(enemy_table[i].attack==0 &&
     enemy_table[i].attack_ticks<=0 &&
     enemy_table[i].ship.sp_id!=FUEL_SP_ID){

     if (rand()%2==1) {enemy_table[i].attack=1;enemy_table[i].ship.sp_pid=ENEMY_ATTACK_SP_PID;}
   }



   if(enemy_table[i].attack_ticks<=0){enemy_table[i].attack_ticks=ENEMY_ATTACK_TICKS;}



   if (enemy_table[i].ship.posX>0 &&
      enemy_table[i].ship.posX<SCREEN_MAX_X-SP_W &&
      enemy_table[i].ship.posY>0 &&
      enemy_table[i].ship.posY<SCREEN_MAX_Y-SP_L &&
      enemy_table[i].attack){

      dx = pl->ship.posX - enemy_table[i].ship.posX;
      dy = pl->ship.posY - enemy_table[i].ship.posY;
      if (dx>ENEMY_SP_SPEED)

      dx=ENEMY_SP_SPEED;

      else if(dx<-ENEMY_SP_SPEED)

      dx=-ENEMY_SP_SPEED;




      if (dy>ENEMY_SP_SPEED||dy==0)

      dy=ENEMY_SP_SPEED;

      else if(dy<-ENEMY_SP_SPEED)

      dy=-ENEMY_SP_SPEED;


      int miss = rand()%6; //miss tracking


      if(miss>3) enemy_table[i].ship.vx=dx;


      if(miss>2) enemy_table[i].ship.vy=dy;


      }





   if (enemy_table[i].ship.posX>=GAME_MAX_X) {enemy_table[i].ship.posX=GAME_MAX_X;
 enemy_table[i].ship.vx=-ENEMY_SP_SPEED;}

   else if (enemy_table[i].ship.posX<=GAME_MIN_X){enemy_table[i].ship.posX=GAME_MIN_X;
 enemy_table[i].ship.vx=ENEMY_SP_SPEED;};


   if (enemy_table[i].ship.posY>=GAME_MAX_Y) {enemy_table[i].ship.posY=GAME_MAX_Y;
 enemy_table[i].ship.vy=-ENEMY_SP_SPEED;}

   else if (enemy_table[i].ship.posY<=GAME_MIN_Y)
 {enemy_table[i].ship.posY=GAME_MIN_Y;enemy_table[i].ship.vy=ENEMY_SP_SPEED;}
```

```
  enemy_table[i].ship.posX+=enemy_table[i].ship.vx;

  enemy_table[i].ship.posY+=enemy_table[i].ship.vy;


  //blink/toggle fuel sprite color

  enemy_table[i].ticks++; enemy_table[i].ticks&=0x1F;

  if(enemy_table[i].ticks==0) enemy_table[i].toggle^=1;

  if(enemy_table[i].ship.sp_id==FUEL_SP_ID)enemy_table[i].ship.sp_pid =
ENEMY_SP_PID+enemy_table[i].toggle;



 }



}


/*

* Update Sprite Map

*

*

*

*

*/



void rr_spriteMap_update(rr_game_t*gm, rr_player_t *pl){


 //player sprites
```

```
/*// animate explosion before stopping game
  if(pl->ship.die_ticks>0){
    pl->ship.die_ticks--;
    if(pl->ship.die_ticks==EXPLODE_TICKS/2) {pl->ship.sp_id=ENEMY_COLSN2_SP_ID;}


    if(pl->ship.die_ticks<=0){
      gm->play=0;
      pl->ship.posX=GAME_MIN_X;
      pl->ship.posY=GAME_MIN_Y;
    }
   }*/


 set_spriteMAP(gm,
            pl->ship.sp_num,
            pl->ship.sp_id,
            pl->ship.sp_pid,
            pl->ship.posX,
            pl->ship.posY);


   //enemy sprites
  for(int i=0; i<GAME_MAX_ENEMY; i++){


   if(enemy_table[i].ship.alive==0) {
    continue;
   } else if(enemy_table[i].ship.die_ticks>0){
    enemy_table[i].ship.die_ticks--;
    if(enemy_table[i].ship.die_ticks==EXPLODE_TICKS/2)
{enemy_table[i].ship.sp_id=ENEMY_COLSN2_SP_ID;}
    //printf("Die Ticks:%d\n", enemy_table[i].ship.die_ticks);//DEBUG
    if(enemy_table[i].ship.die_ticks<=0){
      enemy_table[i].ship.alive=0;
      enemy_table[i].ship.posX=GAME_MIN_X;
      enemy_table[i].ship.posY=GAME_MIN_Y;
      gm->score++;

      #ifdef DEBUG_PRINT
      printf("Enemy:%d removed\n",i); //DEBUG
      #endif

    }

   }



    set_spriteMAP(gm,
              i+ENEMY_SP_NUM_START,
             enemy_table[i].ship.sp_id,
             enemy_table[i].ship.sp_pid,
             enemy_table[i].ship.posX,
             enemy_table[i].ship.posY);
  }



   //missile sprites
   for(int i=0; i<PLAYER_MAX_MISSILE;i++){

    if(missile[i].ship.alive==0){
      continue;
```

```
      }else if(missile[i].ship.die_ticks>0){
         missile[i].ship.die_ticks--;
       if (missile[i].ship.die_ticks<=0){
          missile[i].ship.alive=0;
          missile[i].ship.posX=GAME_MIN_X;
          missile[i].ship.posY=GAME_MIN_Y;
       }
     }

     set_spriteMAP(gm,
                    missile[i].ship.sp_num,
                    missile[i].ship.sp_id,
                    missile[i].ship.sp_pid,
                    missile[i].ship.posX,
                    missile[i].ship.posY);

    }

    fuel_sprite_update(gm, pl);
    // fuel_update(&player);
    score_sprite_update(gm);

    if(gm->reset_dash)gm->reset_dash=0; //reset dashboard after reset

}

/*
* Collision detection
* Note: Collision could be detected during enemy_update or player_update
```

```
*

*

*

*/



void rr_collision_detect(rr_game_t*gm, rr_player_t *pl){


  for(int i=0; i<GAME_MAX_ENEMY; i++){

    if(enemy_table[i].ship.alive==0 || enemy_table[i].ship.die_ticks>0) continue;

 //player-enemy collision detection

  if (((enemy_table[i].ship.posX + SP_W >= pl->ship.posX)&&

      (enemy_table[i].ship.posX<=pl->ship.posX + SP_W)&&

      (enemy_table[i].ship.posY + SP_L >= pl->ship.posY)&&

      (enemy_table[i].ship.posY<=pl->ship.posY + SP_L)))

      {


       //TODO: crash player ship

        if(enemy_table[i].ship.sp_id==FUEL_SP_ID) {

          enemy_table[i].ship.die_ticks=1;

          if(pl->fuel<PLAYER_MAX_FUEL) {

            pl->fuel = pl->fuel + FUEL_REWARD > PLAYER_MAX_FUEL ? PLAYER_MAX_FUEL : pl->fuel +
FUEL_REWARD;

            pl->fuel_ticks = PLAYER_FUEL_TICKS;


          }

          set_audio(gm, AUDIO_FUEL);

          #ifdef DEBUG_PRINT

          printf("Fuel++\n");//DEBUG
          #endif


        }else{
             //crash enemy ship
             player_stop_game(gm,pl);
        }


      }



     for(int j=0; j<PLAYER_MAX_MISSILE;j++){


    if(missile[j].ship.alive==0) continue;


      //missile-enemy collision detection
      if (((enemy_table[i].ship.posX + SP_W >= missile[j].ship.posX)&&
            (enemy_table[i].ship.posX<=missile[j].ship.posX + SP_W)&&
            (enemy_table[i].ship.posY + SP_L >= missile[j].ship.posY)&&
            (enemy_table[i].ship.posY<=missile[j].ship.posY + SP_L)))
            {
             #ifdef DEBUG_PRINT
             printf("Missile Strike on enemy#:%d\n",j);//DEBUG
             #endif


             set_audio(gm, AUDIO_EXPLODE);


             //destroy missile. update during rr_player_update
             //missile[j].ship.alive=0;
             missile[j].ship.die_ticks=1;
```

```
            missile[j].ship.posX=GAME_MIN_X;
            missile[j].ship.posX=GAME_MIN_Y;

            //crash enemy ship
            enemy_table[i].ship.sp_id = ENEMY_COLSN_SP_ID;
            enemy_table[i].ship.die_ticks = EXPLODE_TICKS;

          }
   }

  }



}

/*
 * Fuel Sprite Update
 *
 */
static int PREV_FUEL = -1;

void fuel_sprite_update(rr_game_t*gm, rr_player_t *pl) {
  if (PREV_FUEL == pl->fuel && !gm->reset_dash) return;
  else PREV_FUEL = pl->fuel;

  #ifdef DEBUG_PRINT
  printf("Fuel Level: %d\n", pl->fuel); //DEBUG
  #endif
```

```
uint8_t sp_num = FUEL_SP_NUM_END;

// The sign
set_spriteMAP(gm,
        sp_num--,
        FUEL_SIGN_SP_ID,
        FUEL_PAL_ID,
        FUEL_SIGN_X,
        FUEL_SIGN_Y
);

if (pl->fuel >= 10) {
  // The digit 1
  set_spriteMAP(gm,
          sp_num--,
          DIGIT_O_SP_ID + 1,
          FUEL_PAL_ID,
          FUEL_FIRST_DIGIT_X,
          FUEL_SIGN_Y
  );
  // The digit 0
  set_spriteMAP(gm,
          sp_num--,
          DIGIT_O_SP_ID,
          FUEL_PAL_ID,
          FUEL_SECOND_DIGIT_X,
          FUEL_SIGN_Y
  );
```

```
 }
 else if (pl->fuel < 10) {
  // The first digit
  set_spriteMAP(gm,
        sp_num--,
        DIGIT_O_SP_ID,
        FUEL_PAL_ID,
        FUEL_FIRST_DIGIT_X,
        FUEL_SIGN_Y
  );

  // The first digit
  set_spriteMAP(gm,
        sp_num--,
        DIGIT_O_SP_ID + pl->fuel,
        FUEL_PAL_ID,
        FUEL_SECOND_DIGIT_X,
        FUEL_SIGN_Y
  );
 }
}

/*
 * Fuel Update
 *
 *
int FUEL_TICKS = 0;
void fuel_update(rr_player_t *pl) {
 if (pl->fuel <= 0) return;
 if (FUEL_TICKS < FUEL_DEC_TICK) {
  FUEL_TICKS++;
 }
 else {
  FUEL_TICKS = 0;
  pl->fuel--;
 }
}*/



/*
 * Score Sprite Update
 */
static int PREV_SCORE = -1;

void score_sprite_update(rr_game_t*gm) {
 if (PREV_SCORE == gm->score && !gm->reset_dash) return;
 else PREV_SCORE = gm->score;

 uint8_t sp_num = SCORE_SP_NUM_END;

 #ifdef DEBUG_PRINT
 printf("Score Point: %d\n",gm->score);//DEBUG
 #endif

 // The sign
 set_spriteMAP(gm,
        sp_num--,
            SCORE_SIGN_SP_ID,
            SCORE_PAL_ID,
            SCORE_SIGN_X,
            SCORE_SIGN_Y
 );

 int last_digit = (int)(gm->score % 10);

 if (gm->score >= 100) {
   // The first digit
   int first_digit = (int)(gm->score / 100);
   int second_digit = (int)((gm->score % 100) / 10);
   set_spriteMAP(gm,
             sp_num--,
             DIGIT_O_SP_ID + first_digit,
             SCORE_PAL_ID,
             SCORE_FIRST_DIGIT_X,
             SCORE_SIGN_Y
   );
   // The second digit
   set_spriteMAP(gm,
             sp_num--,
             DIGIT_O_SP_ID + second_digit,
             SCORE_PAL_ID,
             SCORE_FIRST_DIGIT_X,
             SCORE_SIGN_Y
   );

   // The third digit
```

```
    set_spriteMAP(gm,
              sp_num--,
              DIGIT_O_SP_ID + last_digit,
              SCORE_PAL_ID,
              SCORE_SECOND_DIGIT_X,
              SCORE_SIGN_Y
    );
  }
  else if (gm->score < 100 && gm->score >= 10) {
    // The first digit
    int first_digit = (int)(gm->score / 10);
    set_spriteMAP(gm,
              sp_num--,
              DIGIT_O_SP_ID + first_digit,
              SCORE_PAL_ID,
              SCORE_FIRST_DIGIT_X,
              SCORE_SIGN_Y
    );

    // The last digit
    set_spriteMAP(gm,
              sp_num--,
              DIGIT_O_SP_ID + last_digit,
              SCORE_PAL_ID,
              SCORE_SECOND_DIGIT_X,
              SCORE_SIGN_Y
    );
  } else if (gm->score < 10) {
    set_spriteMAP(gm,
              sp_num--,
              DIGIT_O_SP_ID + last_digit,
              SCORE_PAL_ID,
              SCORE_FIRST_DIGIT_X,
              SCORE_SIGN_Y
    );
  }
 }

 /*
 * rr_read_hw_stat
 * Reads hardware status
 *
 *
 *
 *
 *
 */
 void rr_read_hw_stat(rr_game_t*gm){
  status_t status;

  read_status(gm, &status); //GPIO status
  gm->hw_stat = status.raw32;

 }

 /*
 *  player_stop_game()
 *
```

```
*
*
*
*
*
*/
static void player_stop_game( rr_game_t *gm, rr_player_t *pl){
  pl->ship.sp_id = ENEMY_COLSN_SP_ID;
  pl->ship.sp_pid = PLAYER_SP_PID;
  pl->ship.die_ticks = EXPLODE_TICKS;
  gm->play=0;   //stop game
  gm->start_ticks=50;
  set_audio(gm, AUDIO_EXPLODE);
}

/*
* rr_is_fbtn_pressed
* Check to see if fire button is pressed
*
*
*
*
*
*/

int rr_is_fbtn_pressed(rr_game_t*gm)

{
  int ret = -1;
  uint8_t st = 0x1F &gm->hw_stat;


  if(!(st & JOYSTICK_FIRE)) ret = 1;


  return ret;
}


/*
* rr_is_frame_update
* Check to see if frame update bit is high
*
*
*
*
*
*/

int rr_is_frame_update(rr_game_t*gm)

{

  int ret = -1;
  uint8_t st = 0x1F &gm->hw_stat;
  if(!(st & UPDATE_FRAME)) ret = 1;

  return ret;
}
```

vga_ball.c

```
/* module:vga_ball.c
 * Team - River Raid
 * Rojan Banmali (rb3199)
 * Xinhao Su (xs2413)
 *
 * CSEE4840
 * Stephen A. Edwards
 * Columbia University
 */


/* * Device driver for the River Raid
 *
 * A Platform device implemented using the misc subsystem
 *
 * Stephen A. Edwards
 * Columbia University
 *
 * References:
 * Linux source: Documentation/driver-model/platform.txt
 *              drivers/misc/arm-charlcd.c
 * http://www.linuxforu.com/tag/linux-device-drivers/
 * http://free-electrons.com/docs/
 *
 * "make" to build
 * insmod vga_ball.ko
 *
 * Check code style with
 * checkpatch.pl --file --no-tree vga_ball.c
 */

#include <linux/module.h>
#include <linux/init.h>
#include <linux/errno.h>
#include <linux/version.h>
#include <linux/kernel.h>
#include <linux/platform_device.h>
#include <linux/miscdevice.h>
#include <linux/slab.h>
#include <linux/io.h>
#include <linux/of.h>
#include <linux/of_address.h>
#include <linux/fs.h>
#include <linux/uaccess.h>
#include "vga_ball.h"

#define DRIVER_NAME "vga_ball"

/* Device registers */
#define DEV_TILE_MAP(x)         (x)
#define DEV_SPRITE_MAP(x)       ((x)+4)
#define DEV_AUDIO(x)            ((x)+8)
#define DEV_VSCROLL(x)                  ((x)+12)
#define DEV_STATUS(x)           ((x)+16)
```

```
/*
 * Information about our device
 */
struct vga_ball_dev {
        struct resource res; /* Resource: our registers */
        void __iomem *virtbase; /* Where registers can be accessed in memory */
     tileMAP_t  tileMAP;
                spriteMAP_t     spriteMAP;
                audio_t         audio;
                vscroll_t       vscroll;
                status_t status;
} dev;

/*
 * Write segments of a single digit
 * Assumes digit is in range and the device information has been set up
 */
static void write_tileMAP(tileMAP_t *ptm)
{
        iowrite32(ptm->raw32, DEV_TILE_MAP(dev.virtbase));
        dev.tileMAP = *ptm;
}

static void write_spriteMAP(spriteMAP_t *psm)
{
        iowrite32(psm->raw32, DEV_SPRITE_MAP(dev.virtbase));
        dev.spriteMAP = *psm;
}
```

```
static void write_audio(audio_t *pa)
{
	iowrite32(pa->raw32, DEV_AUDIO(dev.virtbase));
	dev.audio = *pa;
}

static void write_vscroll(vscroll_t *pvs)
{
	iowrite32(pvs->raw32, DEV_VSCROLL(dev.virtbase));
	dev.vscroll = *pvs;
}

static void read_status(void)
{
	dev.status.raw32 = ioread32(DEV_STATUS(dev.virtbase));
}

/*
 * Handle ioctl() calls from userspace:
 * Read or write the segments on single digits.
 * Note extensive error checking of arguments
 */
static long vga_ball_ioctl(struct file *f, unsigned int cmd, unsigned long arg)
{
	vga_ball_arg_t vla;

	switch (cmd) {
	case VGA_BALL_WRITE_TILE_MAP:
    if (copy_from_user(&vla, (vga_ball_arg_t *) arg,
                sizeof(vga_ball_arg_t)))
        return -EACCES;
    write_tileMAP(&vla.tileMAP);
    break;

case VGA_BALL_WRITE_SPRITE_MAP:{
if (copy_from_user(&vla, (vga_ball_arg_t *) arg,
                sizeof(vga_ball_arg_t)))
        return -EACCES;
    write_spriteMAP(&vla.spriteMAP);

    break;
}

    case VGA_BALL_WRITE_AUDIO:{
    if (copy_from_user(&vla, (vga_ball_arg_t *) arg,
                sizeof(vga_ball_arg_t)))
        return -EACCES;
    write_audio(&vla.audio);

    break;
}

case VGA_BALL_WRITE_VSCROLL:{

if (copy_from_user(&vla, (vga_ball_arg_t *) arg,
                sizeof(vga_ball_arg_t)))
        return -EACCES;
```

```
                write_vscroll(&vla.vscroll);



                break;
        }

        case VGA_BALL_READ_STATUS:
                //TODO: read from hardware
                  read_status();
                   vla.vscroll = dev.vscroll;
                   vla.status = dev.status;
                if (copy_to_user((vga_ball_arg_t *) arg, &vla,
                                 sizeof(vga_ball_arg_t)))
                        return -EACCES;
                break;

        default:
                return -EINVAL;
        }

        return 0;
}

/* The operations our device knows how to do */
static const struct file_operations vga_ball_fops = {
        .owner          = THIS_MODULE,
        .unlocked_ioctl = vga_ball_ioctl,
};
```

```
/* Information about our device for the "misc" framework -- like a char dev */
static struct miscdevice vga_ball_misc_device = {
	.minor		= MISC_DYNAMIC_MINOR,
	.name		= DRIVER_NAME,
	.fops		= &vga_ball_fops,
};

/*
 * Initialization code: get resources (registers) and display
 * a welcome message
 */
static int __init vga_ball_probe(struct platform_device *pdev)
{
   //vga_ball_color_t beige = {{0xf9,0xe4,0xb7}};
	//TODO: initialize driver

	int ret;

	/* Register ourselves as a misc device: creates /dev/vga_ball */
	ret = misc_register(&vga_ball_misc_device);

	/* Get the address of our registers from the device tree */
	ret = of_address_to_resource(pdev->dev.of_node, 0, &dev.res);
	if (ret) {
		ret = -ENOENT;
		goto out_deregister;
	}

	/* Make sure we can use these registers */
	if (request_mem_region(dev.res.start, resource_size(&dev.res),
			       DRIVER_NAME) == NULL) {
		ret = -EBUSY;
		goto out_deregister;
	}

	/* Arrange access to our registers */
	dev.virtbase = of_iomap(pdev->dev.of_node, 0);
	if (dev.virtbase == NULL) {
		ret = -ENOMEM;
		goto out_release_mem_region;
	}

	/* TODO: Set driver init parameters*/

	return 0;

out_release_mem_region:
	release_mem_region(dev.res.start, resource_size(&dev.res));
out_deregister:
	misc_deregister(&vga_ball_misc_device);
	return ret;
}

/* Clean-up code: release resources */
static int vga_ball_remove(struct platform_device *pdev)
{
	iounmap(dev.virtbase);
	release_mem_region(dev.res.start, resource_size(&dev.res));
	misc_deregister(&vga_ball_misc_device);
	return 0;
}

/* Which "compatible" string(s) to search for in the Device Tree */
#ifdef CONFIG_OF
static const struct of_device_id vga_ball_of_match[] = {
	{ .compatible = "csee4840,vga_ball-1.0" },
	{},
};
MODULE_DEVICE_TABLE(of, vga_ball_of_match);
#endif

/* Information for registering ourselves as a "platform" driver */
static struct platform_driver vga_ball_driver = {
	.driver	= {
		.name	= DRIVER_NAME,
		.owner	= THIS_MODULE,
		.of_match_table = of_match_ptr(vga_ball_of_match),
	},
	.remove		= __exit_p(vga_ball_remove),
};

/* Called when the module is loaded: set things up */
static int __init vga_ball_init(void)
{
	pr_info(DRIVER_NAME ": init\n");
	return platform_driver_probe(&vga_ball_driver, vga_ball_probe);
}
```

```
/* Calball when the module is unloaded: release resources */
static void __exit vga_ball_exit(void)
{
        platform_driver_unregister(&vga_ball_driver);
        pr_info(DRIVER_NAME ": exit\n");
}

module_init(vga_ball_init);
module_exit(vga_ball_exit);

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Stephen A. Edwards, Columbia University");
MODULE_DESCRIPTION("VGA ball driver");
```

utils.c

```
/* module:utils.h
 * Team - River Raid
 * Rojan Banmali (rb3199)
 * Xinhao Su (xs2413)
 *
 * CSEE4840
 * Stephen A. Edwards
 * Columbia University
 */
```

```
#include "rr.h"

/*
* game_init
*
*
*
*
*/

int game_init(rr_game_t *gm, game_thread_fp gfn, const char *drv_fn,  const char *tile_fn){

  //open driver file
  printf("Game Userspace program started\n");

  if ( (gm->driver_fd = open(drv_fn, O_RDWR)) == -1) {
    fprintf(stderr, "could not open %s\n", drv_fn);
    perror("game_init driver fail\n");
    exit(1);
  }

  printf("driver_fd:%d\n",gm->driver_fd);

  //open tile map file
  FILE *fp = fopen(tile_fn, "r");

  if(fp == NULL)
```

```
 {
    printf("Error opening tileMAP file\n");
    exit(1);
 }
 gm->tile_fp=fp;

 //set terminal
 gm->term_old = (struct termios) {0};

 if (tcgetattr(0, &(gm->term_old)) < 0)
 perror("tcsetattr()");

 gm->term_old.c_lflag &= ~ICANON;
 gm->term_old.c_lflag &= ~ECHO;
 gm->term_old.c_cc[VMIN] = 1;
 gm->term_old.c_cc[VTIME] = 0;

 if (tcsetattr(0, TCSANOW, &gm->term_old) < 0)
 perror("tcsetattr ICANON");

 gm->play=0;
 //start game thread
 pthread_create(&(gm->thread), NULL, gfn, NULL);



 return 0;
}

/*
```

```
* game_init
*
*
*
*
*/

void game_close(rr_game_t *gm){

  set_audio(gm, AUDIO_OFF);

  fclose(gm->tile_fp);
  pthread_cancel(gm->thread);
  pthread_join(gm->thread, NULL);

  //set to orignal terminal flag
  gm->term_old.c_lflag |= ICANON;
  gm->term_old.c_lflag |= ECHO;
  if (tcsetattr(0, TCSADRAIN, &gm->term_old) < 0) perror ("tcsetattr ~ICANON");

}



/*
* read_status
*
*
*
*
```

```
*/

int read_status(rr_game_t *gm, status_t *st) {

  vga_ball_arg_t vla;

  if (ioctl(gm->driver_fd, VGA_BALL_READ_STATUS, &vla)) {
      perror("ioctl(VGA_BALL_READ_STATUS) failed");
      return -1;
  }

  *st = vla.status;

  return  0;

}

/*
* print_status
*
*
*
*
*/
void print_status(rr_game_t *gm) {

  vga_ball_arg_t vla;

  if (ioctl(gm->driver_fd, VGA_BALL_READ_STATUS, &vla)) {
    perror("ioctl(VGA_BALL_READ_STATUS) failed");
    return;
  }

  printf("status: %d, %d, %d, %d, %d, %d\n",
      vla.status.bits.joystick_UP,
      vla.status.bits.joystick_DOWN,
      vla.status.bits.joystick_LEFT,
      vla.status.bits.joystick_RIGHT,
      vla.status.bits.joystick_FIRE,
      vla.status.bits.update_frame);
     // printf("status: %d\n",vla.status.raw32);
}



/*
* set_vscroll
*
*
*
*
*/
/* Set the background color */
void set_vscroll(rr_game_t *gm, vscroll_t *scroll)
{
  vga_ball_arg_t vla;
  vla.vscroll = *scroll;
  if (ioctl(gm->driver_fd, VGA_BALL_WRITE_VSCROLL, &vla)) {
    printf("driver_fd:%d\n",gm->driver_fd);
```

```
     perror("ioctl(VGA_BALL_WRITE_VSCROLL) failed:%d");
     return;
  }
}

/*
* set_tileMAP
*
*
*
*
*/
/* Set the tile MAP color */
void set_tileMAP(rr_game_t *gm, tileMAP_t *tm)
{
  vga_ball_arg_t vla;
  vla.tileMAP = *tm;
  if (ioctl(gm->driver_fd, VGA_BALL_WRITE_TILE_MAP, &vla)) {
     perror("ioctl(VGA_BALL_WRITE_TILE_MAP) failed");
     return;
  }
}

/*
* set_audio
*
*
*
*
```

```
*/
/* Set the background color */
void set_audio(rr_game_t *gm, uint8_t cmd)
{
  vga_ball_arg_t vla;
  vla.audio.raw32 = (uint32_t)cmd;
  if (ioctl(gm->driver_fd, VGA_BALL_WRITE_AUDIO, &vla)) {
      perror("ioctl(VGA_BALL_WRITE_AUDIO) failed");
      return;
  }
}

/*
* set_spriteMAP
*
*
*
*
*/
/* Set the tile MAP color */
void set_spriteMAP(rr_game_t *gm, uint8_t spNum, uint8_t spID, uint8_t palID, int16_t x, int16_t y)
{

  vga_ball_arg_t vla;
  spriteMAP_t sm;

  sm.raw32 = ((spNum & 0x1F)<<26)|
            ((0x3FF & x)<<16) |
            ((0x1FF & y) <<7) |
        (spID<<2) |
        (palID & 0x03);

  vla.spriteMAP = sm;
  if (ioctl(gm->driver_fd, VGA_BALL_WRITE_SPRITE_MAP, &vla)) {
    perror("ioctl(VGA_BALL_WRITE_SPRITE_MAP) failed");
    return;
  }
}
```

main.c

```
/* module:main.c
 * Team - River Raid
 * Rojan Banmali (rb3199)
 * Xinhao Su (xs2413)
 *
 * CSEE4840
 * Stephen A. Edwards
 * Columbia University
 */

#include "rr.h"

//objects
rr_player_t player;
rr_game_t game;
```

```
rr_tile_t tile;


void *game_thread_f(void *);


struct timespec ts_t1, ts_t2;

long  del_ms, elapsed_ms;


/*

* C main program

* Start game thread and handle keyboard,etc.

*

*

*

*

*/


int main()

{


  game_init(&game, &game_thread_f,  "/dev/vga_ball", "screen-arrange.txt");


  printf("Type 'q' and return to quit\n");


  //set echo off

  char buf = 0;


    clock_gettime(CLOCK_MONOTONIC,&ts_t1);

    clock_gettime(CLOCK_MONOTONIC,&ts_t2);
  while (buf!='q'){


    clock_gettime(CLOCK_MONOTONIC,&ts_t2);
    del_ms = (ts_t2.tv_sec-ts_t1.tv_sec)*1000000 + (ts_t2.tv_nsec-ts_t1.tv_nsec)/1000;
    elapsed_ms+=del_ms;
    ts_t1=ts_t2;


    //FOR TESTING
    if (read(0, &buf, 1) < 0) perror ("read()");


    switch(buf){


      case 'k':{ //LEFT


        break;
      }


      case ';':{ //RIGHT


        break;
      }
      case 'l':{ //DOWN


        break;
      }


      case 'o':{ //UP


        break;
        }

      }

     }

  game_close(&game);

  return 0;
}

/*
* Main Game loop.
*
*
*
*
*/

void *game_thread_f(void *ignored)
{


  rr_tile_init(&game, &tile);


  while (1) {

    rr_read_hw_stat(&game);
    if(game.play){

      rr_player_update(&game, &player);
      rr_enemy_update(&game, &player);
      rr_collision_detect(&game, &player);
      if(rr_is_frame_update(&game)>0) {
        rr_spriteMap_update(&game, &player);
        rr_tile_update(&game, &tile);
      }

    }else{

      if (rr_is_fbtn_pressed(&game)>0 && game.start_ticks<=0) { //wait for a few ticks
        game.play=1;
        printf("Begin Mission!\n");//DEBUG
        rr_reset(&game, &player);
      }else if(game.start_ticks>0){
        game.start_ticks--;
        if (game.start_ticks<=0) set_audio(&game, AUDIO_OFF);

      }

    }

    usleep(GAME_POLL_WAIT); //in usec
  }

  return NULL;
```

```
}
```

rr.h

```
/* module:rr.h
 * Team - River Raid
 * Rojan Banmali (rb3199)
 * Xinhao Su (xs2413)
 *
 * CSEE4840
 * Stephen A. Edwards
 * Columbia University
 */

#ifndef _RR_H
#define _RR_H

#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <time.h>
#include "vga_ball.h"
#include <sys/ioctl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
```

```
#include <string.h>
#include <unistd.h>
#include <pthread.h>
#include <termios.h>
#include <time.h>

#define FPS 10
#define REF_PREIOD ((1/FPS)*1000000)

#define SCREEN_MAX_X 320
#define SCREEN_MAX_Y 240
#define SCREEN_MIN_X 0
#define SCREEN_MIN_Y 0

#define GAME_MAX_X 320+16
#define GAME_MAX_Y 240+16
#define GAME_MIN_X 0-18
#define GAME_MIN_Y 0-18
#define GAME_MAX_ENEMY 7
#define GAME_POLL_WAIT 2*5000


#define SP_W 16
#define SP_L 16

#define EXPLODE_TICKS 30

#define ENEMY_SP_ID_START 2
#define ENEMY_COLSN_SP_ID 7
```

```
#define ENEMY_COLSN2_SP_ID 6
#define ENEMY_SP_TOTAL_NUM 5
#define ENEMY_SP_NUM_START 0
#define ENEMY_SP_PID 3
#define ENEMY_ATTACK_SP_PID 1
#define ENEMY_SP_SPEED 1
#define ENEMY_ATTACK_TICKS 300
#define FUEL_REWARD 2

#define FUEL_SP_ID 9

#define PLAYER_SP_ID 0
#define PLAYER_COLSN_SP_ID 7
#define PLAYER_SP_NUM 8
#define PLAYER_SP_EMTPY_PID 1
#define PLAYER_SP_PID  2

#define PLAYER_SPEED_X 2
#define PLAYER_SPEED_Y 2
#define PLAYER_MAX_X 320-18
#define PLAYER_MAX_Y 240-16
#define PLAYER_MIN_X 0
#define PLAYER_MIN_Y 2
#define PLAYER_MAX_MISSILE 2
#define PLAYER_MAX_FUEL 10
#define PLAYER_FUEL_TICKS 300
```

```
#define MISSILE_SP_ID 1
#define MISSILE_SP_NUM 7
#define MISSILE_SP_PID 3
#define MISSILE_VY 4

//ReadStatus Mask
#define JOYSTICK_UP     0x01 //UP
#define JOYSTICK_DOWN   0x02 //DOWN
#define JOYSTICK_LEFT   0x04 //LEFT
#define JOYSTICK_RIGHT  0x08 //RIGHT
#define JOYSTICK_FIRE   0x10 //FIRE
#define UPDATE_FRAME    0x20 //update frame

#define AUDIO_SAMPLE_TICKS 35
#define AUDIO_OFF 0
#define AUDIO_FLYING 0b00000001
#define AUDIO_FIRE 0b00000011
#define AUDIO_FUEL 0b00000101
#define AUDIO_EXPLODE 0b00001001



#define DIGIT_O_SP_ID 13

/* Fuel Related */
#define FUEL_SIGN_X 10
#define FUEL_SIGN_Y 24
#define FUEL_SIGN_SP_ID 12
#define FUEL_FIRST_DIGIT_X 25
#define FUEL_SECOND_DIGIT_X 38
```

```
#define FUEL_SP_NUM_END 11

#define FUEL_PAL_ID 3

#define FUEL_DEC_TICK 500


/* Score Related */

#define SCORE_SIGN_X 10

#define SCORE_SIGN_Y 40

#define SCORE_SIGN_SP_ID 11

#define SCORE_FIRST_DIGIT_X 25

#define SCORE_SECOND_DIGIT_X 38

#define SCORE_THIRD_DIGIT_X 51

#define SCORE_SP_NUM_END 15

#define SCORE_PAL_ID 3


//Uncomment to print debug messages

//#define DEBUG_PRINT




typedef void *(*game_thread_fp)(void*);


typedef struct{

   int posX;

   int posY;

   int vx;

   int vy;

   uint8_t sp_num;

   uint8_t sp_id;

   uint8_t sp_pid;

   uint8_t alive;
   int16_t die_ticks;

   uint8_t hitcount;

}rr_ship_t;


typedef struct{

   rr_ship_t ship;

}rr_missile_t;


typedef struct{

   rr_ship_t ship;

   uint8_t fire_pressed;

   int8_t  fuel;

   int16_t fuel_ticks;

}rr_player_t;


typedef struct{

   rr_ship_t ship;

   uint8_t ticks;

   uint8_t toggle;

   uint8_t attack;

   int16_t attack_ticks;


}rr_enemy_t;




typedef struct termios termios_t;


typedef struct{
   int driver_fd; //driver file desc.
   FILE *tile_fp; //tileMap file pointer
      //tile map file data
   pthread_t thread; //thread
   struct termios term_old;
   uint64_t ticks;
   uint8_t  vspd;
   uint32_t add_enemy_ticks;
   int16_t score;
   int audio_ticks;
   uint8_t play;
   int start_ticks;
   uint32_t hw_stat;
   uint8_t reset_dash;
}rr_game_t;


typedef struct{
   vscroll_t vs; //veritcal scroll
   tileMAP_t tm; //tile map
   int32_t addr;
   int32_t line;
   int bound;
}rr_tile_t;


//utils.c prototype
int game_init(rr_game_t *gm, game_thread_fp gfn, const char* drv_fn,  const char* tile_fn);
void game_close(rr_game_t *gm);
int read_status(rr_game_t *gm, status_t *st);
int read_status(rr_game_t *gm, status_t *st);
```

```
void print_status(rr_game_t *gm);
void set_vscroll(rr_game_t *gm, vscroll_t *scroll);
void set_tileMAP(rr_game_t *gm, tileMAP_t *tm);
void set_audio(rr_game_t *gm, uint8_t cmd);
void set_spriteMAP(rr_game_t *gm, uint8_t spNum, uint8_t spID, uint8_t palID, int16_t x, int16_t y);



//rr.c prototype
void rr_reset(rr_game_t *gm, rr_player_t *pl);
void rr_tile_init(rr_game_t*gm, rr_tile_t* tl);
void rr_tile_update(rr_game_t*gm, rr_tile_t* tl);
void rr_player_update(rr_game_t*gm, rr_player_t *pl);

void rr_enemy_update(rr_game_t*gm, rr_player_t*pl);

void rr_spriteMap_update(rr_game_t*gm, rr_player_t *pl);
void rr_collision_detect(rr_game_t*gm, rr_player_t *pl);

int rr_is_fbtn_pressed(rr_game_t*gm);
void rr_read_hw_stat(rr_game_t*gm);
int rr_is_frame_update(rr_game_t*gm);

#endif
```

vga_ball.sv

```
/* module: RiverRaid Hardware
 * Team - River Raid
 * Rojan Banmali (rb3199)
 * Xinhao Su (xs2413)
 *
 *
 * Avalon memory mapped slave
 *
 *
 * CSEE4840
 * Stephen A. Edwards
 * Columbia University
 */

module vga_ball(
   input logic clk,
   input logic reset,
			input logic [31:0]  writedata,
			input logic write,
			input logic read,
   input       chipselect,
			input logic [7:0]  address,
   input logic [7:0] GPIO,
   input left_chan_ready,
			input right_chan_ready,

			output logic [15:0] sample_data_l,
			output logic sample_valid_l,
			output logic [15:0] sample_data_r,
			output logic sample_valid_r,
   output logic [31:0] readdata,
                output logic [7:0] VGA_R, VGA_G, VGA_B,
                output logic    VGA_CLK, VGA_HS, VGA_VS,
                         VGA_BLANK_n,
                output logic        VGA_SYNC_n);

 logic [10:0]      vga_hcount;
 logic [9:0]    vga_vcount;
 logic [8:0]       hcount;
 logic [8:0]    vcount;
 logic [23:0]    vga_rgb;

 logic update_frame, update_vars;

 logic [8:0] tileRAM_addr;    //tile MAP rom address
 logic [8:0] tileRAM_raddr;   //tile MAP read address
 logic      tileRAM_wren;    //tile MAP write enable
 logic [7:0] tileRAM_data;    //tile MAP read RAM data
 logic [7:0] tileRAM_wdata;   //tile MAP write RAM data

 logic [12:0] tileROM_addr;   //3bit/pixel tile artwork tile ROM address
 logic [4:0] tileROM_pix;     //3bit tile ROM pixel output
 logic [1:0] tile_pal_id;

 //software interface vars
 logic [7:0] vscroll;//, vscroll_;
 logic [2:0] audio_cmd;
 logic audio_on;

 logic [7:0] fg_data; // 3bit unused + 3bit pxl + 2bit palID
```

```
logic [5:0] fg_pix;

logic [1:0] fg_palID;


logic [23:0] sp_rgb_out;

logic [23:0] tile_rgb_out;


//instantiate modules.


tileROM tileROM_1 (.clock(clk), .address(tileROM_addr),.q(tileROM_pix));


tileRAM tileRAM_1 (.clock(clk),

          .address(tileRAM_addr),

          .wren(tileRAM_wren),

          .data(tileRAM_wdata),

          .q(tileRAM_data));



vga_counters counters(.clk50(clk), .hcount(vga_hcount), .vcount(vga_vcount),.*);


audio audio0(.clk(clk),.command(audio_cmd),.*);


sprite_gen sprite_gen0(.clk(clk),

            .reset(reset),

            .chipselect(chipselect),

            .address(address),

            .writedata(writedata),

            .write(write),

            .hcount(hcount),

            .vga_vcount(vga_vcount),
                 .vcount(vcount),

                 .fg_data(fg_data));


 // sprite_pal_rom sprite_pal0(

 //   .palID(fg_palID),

 //   .address(fg_pix),

 //   .rgb_out(sp_rgb_out)

 // );


colorPals colorPal_0(

   .palID(tile_pal_id),

   .address({1'b0, tileROM_pix}),

   .rgb_out(tile_rgb_out)

 );


 colorPals colorPal_1(

   .palID(fg_palID),

   .address(fg_pix),

   .rgb_out(sp_rgb_out)

 );


// Driver Read Write on posedge

  always_ff @(posedge clk) begin


    audio_cmd<=0;


    if (reset) begin

      //reset

       readdata<=0;
       //vscroll_<= 8'h0;

       vscroll<= 8'h0;


    end else if (chipselect && write)begin //WRITE from driver


     case (address)

       8'h0 :begin //Handled Below during negedge

       end


       8'h1:begin //Handled below during negedge

       end


      8'h2 : begin //PlayAudio {[2:0]audioClip_sel}

        audio_on<=writedata[0];

        audio_cmd<=writedata[3:1];

       end


                  8'h3 : begin //WriteVScroll {[7:0]}

        vscroll<=writedata[7:0];

       end


      endcase


    end else if (chipselect && read) begin //READ from driver


      case (address)

       8'h4:begin

         readdata[5:0]<={update_frame, GPIO[4:0]}; //TODO: readdata[5:0] = {update_frame,FIRE,RIGHT,
LEFT, DOWN, UP}
```

```
        end


       endcase
     end
   end



   //Write TileMap from software
   always_ff @(negedge clk) begin


     if (chipselect && write && address==0)begin //WRITE from driver
                    //WriteTileMap {[8:0]tileMap_waddr, [4:0]tileID, [1:0]palID}
          tileRAM_wdata <= writedata[6:0]; //{tileID, palID}
          tileRAM_addr <= writedata[16:7]; //{tileMAP_addr}
          tileRAM_wren<=1;
       end
       else begin
       //reset
        tileRAM_wdata<= 8'h0; //TODO: set to high impedance?
        tileRAM_addr <= tileRAM_raddr;
        tileRAM_wren<=0;
     end

  end

// Update frame: TODO: Not used at the moment. Comeback later.
   always_ff@(posedge clk) begin
     if(reset || (read && address==4 && update_frame))begin
       update_frame<=0;
     end else if(!update_frame && vga_vcount==480)begin
       update_frame<=1;
     end
   end

    /*//Update scrolling during vertical blanking
    always_ff@(posedge clk) begin

      if (reset) begin
        vscroll<=0;
      end
      else if (update_vars) begin
        vscroll<=vscroll_;
      end

    end*/

  //Tile scrolling
  always_comb begin
     if(hcount<320 && vcount<240)
     tileRAM_raddr = (hcount>>4) + (((vcount+vscroll)%256)>>4)*20;
     else tileRAM_raddr=0;
   end

   always_comb begin
    if(hcount<320 && vcount<240)
      tileROM_addr = (tileRAM_data[7:2]<<8) + (hcount%16) + (((vcount+vscroll)%16)<<4);
    else tileROM_addr=0;
   end
```

```
assign fg_pix = fg_data[7:2];
assign fg_palID = fg_data[1:0];

assign tile_pal_id = tileRAM_data[1:0];

//FINAL VIDEO OUTPUT!!
//TODO: Use palatte to assign RGB
//VGA stuff. Draw RGB: 0,0,0 during blanking.
  always_comb begin
     vga_rgb = {8'h0, 8'h0, 8'h0};
    if (VGA_BLANK_n ) begin
       if(fg_pix != 6'd0 && (vga_hcount>1))
          vga_rgb = sp_rgb_out;
       else
          vga_rgb = tile_rgb_out;
   end
  end

assign {VGA_R, VGA_G, VGA_B} = vga_rgb;

assign update_vars = (vga_hcount==0 && vga_vcount == 480); //update anim variables
assign hcount = vga_hcount[10:2]; //divide by 4
assign vcount = vga_vcount[9:1]; //divide by 2

endmodule

/*
* sprite_pal_rom
```

```
*
*
*
*
*/
// module sprite_pal_rom (
//   input logic [1:0] palID,
//   input logic [2:0] address,
//   output logic [23:0] rgb_out
// );

// always_comb begin

//   rgb_out = 24'd0;
// if(palID==0)begin //ship
//   case(address)
//     0: rgb_out = 24'h000000; //transparent
//     1: rgb_out = 24'h888888; //dark gray
//     2: rgb_out = 24'h5B5B5B; //gray light
//     3: rgb_out = 24'h353535; //gray lighter
//     4: rgb_out = 24'h983939; //reddist 1
//     5: rgb_out = 24'h9B190A; //reddist 2
//     6: rgb_out = 24'hE84AFA; //magentaish
//     7: rgb_out = 24'hFFFFFF; //white

//   endcase
// end

// else if (palID==1)begin //explosion
//  case(address)
//     0: rgb_out = 24'h000000; //transparent
//     1: rgb_out = 24'h592A0A; //
//     2: rgb_out = 24'h6A370F; //
//     3: rgb_out = 24'h91601B; //
//     4: rgb_out = 24'hA88524; //
//     5: rgb_out = 24'h5A4C21; //
//     6: rgb_out = 24'h837927; //
//     7: rgb_out = 24'hB0B549; //
//   endcase

// end

// else if (palID==2)begin //TODO: modify and use for tiles
//  case(address)
//     0: rgb_out = 24'h000000; //transparent
//     1: rgb_out = 24'h888888; //dark gray
//     2: rgb_out = 24'h5B5B5B; //gray light
//     3: rgb_out = 24'h353535; //gray lighter
//     4: rgb_out = 24'h983939; //reddist 1
//     5: rgb_out = 24'h9B190A; //reddist 2
//     6: rgb_out = 24'hE84AFA; //magentaish
//     7: rgb_out = 24'hFFFFFF; //white

//   endcase

// end
// else if(palID==3)begin //TODO: modify and use for tiles
//  case(address)
//     0: rgb_out = 24'h000000; //transparent
//     1: rgb_out = 24'h888888; //dark gray
//     2: rgb_out = 24'h5B5B5B; //gray light
//     3: rgb_out = 24'h353535; //gray lighter
//     4: rgb_out = 24'h983939; //reddist 1
//     5: rgb_out = 24'h9B190A; //reddist 2
//     6: rgb_out = 24'hE84AFA; //magentaish
//     7: rgb_out = 24'hFFFFFF; //white

//   endcase
// end
// end

// endmodule



/*
* sprite_gen module
*
*
*
*
*
*/

module sprite_gen(
   input   logic        clk,
   input   logic         reset,

   input                 chipselect,

   input   logic [7:0]   address,

                         input   logic [31:0]  writedata,

                         input   logic         write,

   input   logic [8:0]          hcount,

   input   logic [9:0]          vga_vcount,

   input   logic [8:0]   vcount,


                         output  logic [7:0]  fg_data);




   logic [4:0]  spriteRAM_addr;   //sprite MAP address (for 16 sprites)

   logic [4:0]  spriteRAM_raddr;  //sprite MAP read address (for 16 sprites)

   logic [4:0]  spriteRAM_waddr;


   logic        spriteRAM_wren;   //sprite MAP write enable

   logic [25:0] spriteRAM_data;   //sprite MAP read RAM data

   logic [25:0] spriteRAM_wdata;  //sprite MAP write RAM data


   logic [12:0] spriteROM_addr;   //3bit/pixel tile artwork tile ROM address


   logic [5:0]  spriteROM_pix;    //3bit sprite ROM pixel output


   logic [8:0] row0_addr, row1_addr, row_waddr,row_raddr;

   logic [8:0] row0_addr_b, row1_addr_b;

   logic       row0_wren, row1_wren, row_wren;

   logic       row0_wren_b,row1_wren_b;

   logic [7:0] row0_data,row1_data, row0_wdata,row1_wdata;
```

```
  logic drawpix;

  logic [8:0] row_next;


  //TODO:check width

  logic signed [8:0] sp_posY;

  logic signed [9:0] sp_posX;


  logic signed [8:0] sp_dy;

  logic signed [9:0] sp_dx;


  logic [1:0] sp_palID;

  logic [4:0] sp_ID;


  logic [3:0] spID_ptr;


 //Write sprite Map from software

 always_ff @(negedge clk) begin


   if (chipselect && write && address==1)begin //WRITE from driver

            //WriteSpriteMap {[3:0]spriteMap_waddr, PosX[9:0], PosY[8:0], [4:0]spriteID, [1:0]palID}

     spriteRAM_wdata <= writedata [25:0]; //{PosX, PosY, spriteID, palID}

     spriteRAM_waddr <= writedata  [30:26]; //{spriteMAP_addr - 5bits}

     spriteRAM_wren<=1;

    end

    else begin

    //reset

     spriteRAM_wdata<= 30'h0;

     spriteRAM_waddr<= spriteRAM_raddr;

     spriteRAM_wren<=0;
    end


  end
  //sprite ram read or write operation
  always_comb begin
    if(spriteRAM_wren)
     spriteRAM_addr =  spriteRAM_waddr;
     else
     spriteRAM_addr = spriteRAM_raddr;
  end


    //sprite gen state machine


    enum {
        IDLE,       // await until hcount==0
        READ_WAIT,  // wait until drawpix
        DRAW        // draw
    } state;



//Row to write
  always_comb begin
    if(vcount==262) //end of vcount after it wraps up to 0
     row_next=0;
      else
     row_next=vcount+1;
  end


//Sprite generator FSM:
    always_ff @(negedge clk) begin
     if (reset)begin
       state<=IDLE;
       spriteRAM_raddr<=0; //TODO: For testing only.
     end
     else

     case(state)

      IDLE:begin
        spriteRAM_raddr<=0;
        spID_ptr<=0;
        sp_dx<=0;
        if(hcount==0) state<=READ_WAIT;

      end

      READ_WAIT:begin

     if(drawpix)begin
        state<=DRAW;
      end else begin
        if (spriteRAM_raddr==15)
        state <=IDLE;
        else spriteRAM_raddr<=spriteRAM_raddr+1;
      end

      end
      DRAW:begin

        sp_dx<=sp_dx+1 ;

        if(sp_dx==15)begin
         sp_dx<=0;
         if (spriteRAM_raddr<15)begin
          spriteRAM_raddr<=spriteRAM_raddr+1;
          state<=READ_WAIT;
         end
         else
          state<=IDLE;

        end

      end

     endcase

    end

    //mux: select row0 or row1
    always_comb begin

     row0_wdata=0; row1_wdata=0;
     row1_wren = 0;  row0_wren = 0;
     row0_wren_b=0;row1_wren_b=0;
     //write to row 0
     //read from row 1
      if(vcount[0])begin
        row0_addr = row_waddr;
        row0_wren = row_wren && (spriteROM_pix!=0);
        row0_wdata = {spriteROM_pix,sp_palID[1:0]}; //3(unused)+3+2;
        row1_addr = row_raddr;
        fg_data = row1_data; //3bit final foreground pix. TODO: include palID in row
        if(vga_vcount[0]) row1_wren_b = 1;
      end
     //write to row 1
     //read from row 0
      else begin
        row1_addr = row_waddr;
        row1_wren = row_wren && (spriteROM_pix!=0);
        row1_wdata = {spriteROM_pix,sp_palID[1:0]}; //3(unused)+3+2;
        row0_addr = row_raddr;
        fg_data = row0_data;    //3bit final foreground pix. TODO: include palID in row
        if(vga_vcount[0]) row0_wren_b = 1;


      end


    end


    assign row0_addr_b = row0_addr -1;
    assign row1_addr_b = row1_addr -1;


    //Wrap around
    always_comb begin
     if (hcount<320)
```

```
    row_raddr = hcount;
    else row_raddr = 0;
   end

  //One ff delay to match lines
 //assign row_waddr = sp_posX + sp_dx;
 //assign row_wren =  (state==DRAW && drawpix)? 1:0;
  always_ff @(negedge clk)begin
    row_waddr <= sp_posX + sp_dx;
    row_wren  <=  (state==DRAW && drawpix)? 1:0;
  end

 assign {sp_posX[9:0],sp_posY[8:0],sp_ID[4:0],sp_palID[1:0]} = spriteRAM_data;
 assign sp_dy = (row_next - sp_posY);
 assign spriteROM_addr = (sp_ID<<8) + sp_dx + (sp_dy<<4);
 assign drawpix = (sp_dy>=0 && sp_dy<15 && sp_dx<16);



   rowRAM2 row0(  .clock(clk),
                         .data_a(row0_wdata),
               .address_a(row0_addr),
               .wren_a(row0_wren),
                         .q_a(row0_data),
               .data_b(8'd0),
               .address_b(row0_addr_b),
               .wren_b(row0_wren_b),
                         .q_b()
                         );
  rowRAM2 row1(  .clock(clk),
                    .data_a(row1_wdata),
            .address_a(row1_addr),
            .wren_a(row1_wren),
                    .q_a(row1_data),
            .data_b(8'd0),
            .address_b(row1_addr_b),
            .wren_b(row1_wren_b),
                    .q_b()
                    );



  spriteROM spriteROM_1 (.clock(~clk),.address(spriteROM_addr),.q(spriteROM_pix));


  spriteRAM spriteRAM_1 (.clock(clk),
                .address(spriteRAM_addr),
                .wren(spriteRAM_wren),
                .data(spriteRAM_wdata),
                .q(spriteRAM_data));



endmodule




/*
* vga_counters module:
* generates video signal
```

```
*
*
*/
module vga_counters(
 input logic            clk50, reset,
 output logic [10:0] hcount,  // hcount[10:1] is pixel column
 output logic [9:0]  vcount,  // vcount[9:0] is pixel row
 output logic          VGA_CLK, VGA_HS, VGA_VS, VGA_BLANK_n, VGA_SYNC_n);

/*
 * 640 X 480 VGA timing for a 50 MHz clock: one pixel every other cycle
 *
 * HCOUNT 1599 0            1279       1599 0
 *            ________________             _________
 * ___________|    Video      |_____________|  Video
 *
 *
 * |SYNC| BP |<-- HACTIVE -->|FP|SYNC| BP |<-- HACTIVE
 *       _________________________     ______________
 * |____|       VGA_HS         |____|
 */
  // Parameters for hcount
  parameter HACTIVE      = 11'd 1280,
            HFRONT_PORCH = 11'd 32,
            HSYNC        = 11'd 192,
            HBACK_PORCH  = 11'd 96,
            HTOTAL       = HACTIVE + HFRONT_PORCH + HSYNC +
                           HBACK_PORCH; // 1600
```

```
// Parameters for vcount
parameter VACTIVE      = 10'd 480,
        VFRONT_PORCH = 10'd 10,
        VSYNC        = 10'd 2,
        VBACK_PORCH  = 10'd 33,
        VTOTAL       = VACTIVE + VFRONT_PORCH + VSYNC +
                       VBACK_PORCH; // 525

logic endOfLine;

always_ff @(posedge clk50 or posedge reset)
  if (reset)          hcount <= 0;
  else if (endOfLine) hcount <= 0;
  else                hcount <= hcount + 11'd 1;

assign endOfLine = hcount == HTOTAL - 1;

logic endOfField;

always_ff @(posedge clk50 or posedge reset)
  if (reset)          vcount <= 0;
  else if (endOfLine)
    if (endOfField)   vcount <= 0;
    else              vcount <= vcount + 10'd 1;

assign endOfField = vcount == VTOTAL - 1;

// Horizontal sync: from 0x520 to 0x5DF (0x57F)
// 101 0010 0000 to 101 1101 1111
```

```
  assign VGA_HS = !( (hcount[10:8] == 3'b101) &
                     !(hcount[7:5] == 3'b111));
  assign VGA_VS = !( vcount[9:1] == (VACTIVE + VFRONT_PORCH) / 2);

  assign VGA_SYNC_n = 1'b0; // For putting sync on the green signal; unused

  // Horizontal active: 0 to 1279     Vertical active: 0 to 479
  // 101 0000 0000  1280              01 1110 0000  480
  // 110 0011 1111  1599              10 0000 1100  524
  assign VGA_BLANK_n = !( hcount[10] & (hcount[9] | hcount[8]) ) &
                         !( vcount[9] | (vcount[8:5] == 4'b1111) );

  /* VGA_CLK is 25 MHz
   *           __    __    __
   * clk50   __|  |__|  |__|
   *
   *           _____       __
   * hcount[0]__|     |_____|
   */
  assign VGA_CLK = hcount[0]; // 25 MHz clock: rising edge sensitive

endmodule

/*
* audio module:
*
* uses 8bit audio clips
*/
module audio (
                        input clk, // 50MHz
                        input reset,
  input audio_on, //run audio + flying sound
  input logic [2:0]command, //0:fire, 1:fuel, 2:explode
                        input left_chan_ready,
                        input right_chan_ready,
                        output logic [15:0] sample_data_l,
                        output logic sample_valid_l,
                        output logic [15:0] sample_data_r,
                        output logic sample_valid_r
            );

  parameter DAC_RATE = 50000000/16000;
 //Define audio sample sizes
 //TODO: fill sample sizes appropriately
  parameter AR_FLY_ADDR_ST=0;
  parameter AR_FIRE_ADDR_ST=0;
  parameter AR_FUEL_ADDR_ST=0;
  parameter AR_EXPLODE_ADDR_ST=0;

  parameter AR_FLY_ADDR_END=23213;
  parameter AR_FIRE_ADDR_END=1599;
  parameter AR_FUEL_ADDR_END=2039;
  parameter AR_EXPLODE_ADDR_END=4095;

 logic [11:0] dac_ctr;

 logic dac_ready;
```

```
logic    [15:0]  audio_out;


logic ar_fire_go;
logic ar_fuel_go;
logic ar_explode_go;


logic ar_fly_play;
logic [7:0] ar_fly_q;
logic [15:0] ar_fly_dout;
logic [14:0] ar_fly_rom_addr;


logic ar_fire_play;
logic [7:0] ar_fire_q;
logic [15:0] ar_fire_dout;
logic [10:0] ar_fire_rom_addr;


logic ar_fuel_play;
logic [7:0] ar_fuel_q;
logic [15:0] ar_fuel_dout;
logic [10:0] ar_fuel_rom_addr;


logic ar_explode_play;
logic [7:0] ar_explode_q;
logic [15:0] ar_explode_dout;
logic [11:0] ar_explode_rom_addr;



//TODO: this could be done using a single ROM.
audioROM_fly ar_fly(.address(ar_fly_rom_addr), .clock(clk), .q(ar_fly_q));
```

```
 audioROM_fire ar_fire(.address(ar_fire_rom_addr), .clock(clk), .q(ar_fire_q));

 audioROM_fuel ar_fuel(.address(ar_fuel_rom_addr), .clock(clk), .q(ar_fuel_q));

 audioROM_explode ar_explode(.address(ar_explode_rom_addr), .clock(clk), .q(ar_explode_q));



//Handle audio buffering and also background (flying) sound

 always_ff @(posedge clk) begin


   if(reset || !audio_on) begin

     dac_ctr <=0;

     sample_valid_l <= 0; sample_valid_r <= 0;

     audio_out<=0;

     ar_fly_rom_addr<=AR_FLY_ADDR_ST;

   end


   else if(audio_on && dac_ready && dac_ctr<DAC_RATE) begin


     sample_valid_l <= 0; sample_valid_r <= 0;

     dac_ctr<=dac_ctr+1;


   end else if(audio_on && dac_ready && dac_ctr==DAC_RATE)begin //send DAC output

       dac_ctr<=0;

       ar_fly_rom_addr<=ar_fly_rom_addr+1;


       if(ar_fly_rom_addr==AR_FLY_ADDR_END)begin

           ar_fly_rom_addr<=AR_FLY_ADDR_ST;

       end


       //final output registers
```

```
    audio_out <= ar_fly_dout + ar_fire_dout + ar_fuel_dout + ar_explode_dout;
    sample_valid_l <= 1; sample_valid_r <= 1;

  end

 end


//AUDIO - FIRE
 //fire audio rom
 always_ff @(posedge clk) begin

   if (reset || !audio_on)begin
     ar_fire_play<=0;
   end else if (ar_fire_go) begin
     ar_fire_play<=1;
     ar_fire_rom_addr<=AR_FIRE_ADDR_ST;
   end
   else if (!ar_fire_go && ar_fire_play && dac_ready && dac_ctr==DAC_RATE) begin

     ar_fire_rom_addr<= ar_fire_rom_addr+1;

     if (ar_fire_rom_addr==AR_FIRE_ADDR_END) begin
       ar_fire_play<=0;
     end
   end
 end

 //AUDIO - FUEL
```

```
//fuel audio rom
always_ff @(posedge clk) begin


  if (reset || !audio_on)begin
    ar_fuel_play<=0;
  end else if (ar_fuel_go) begin
    ar_fuel_play<=1;
    ar_fuel_rom_addr<=AR_FUEL_ADDR_ST;
  end
  else if (!ar_fire_go && ar_fuel_play && dac_ready && dac_ctr==DAC_RATE) begin


    ar_fuel_rom_addr<= ar_fuel_rom_addr+1;


    if (ar_fuel_rom_addr==AR_FUEL_ADDR_END) begin
      ar_fuel_play<=0;
    end
  end
end


//AUDIO - EXPLODE
//explode audio rom
always_ff @(posedge clk) begin


  if (reset || !audio_on)begin
    ar_explode_play<=0;
  end else if (ar_explode_go) begin
    ar_explode_play<=1;
    ar_explode_rom_addr<=AR_EXPLODE_ADDR_ST;
  end
```

```
   else if (!ar_fire_go && ar_explode_play && dac_ready && dac_ctr==DAC_RATE) begin
     ar_explode_rom_addr<= ar_explode_rom_addr+1;
     if (ar_explode_rom_addr==AR_EXPLODE_ADDR_END) begin
       ar_explode_play<=0;
     end
   end
 end

 assign ar_fly_dout= {8'h0, ar_fly_q};
 assign ar_fire_dout= {8'h0,ar_fire_q};//(ar_fire_play)? {8'h0,ar_fire_q}:0;
 assign ar_fuel_dout= {8'h0,ar_fuel_q};//(ar_fuel_play)? {8'h0,ar_fuel_q}:0;
 assign ar_explode_dout= {8'h0,ar_explode_q};//(ar_explode_play)? {8'h0,ar_explode_q}:0;

 assign ar_fire_go = command[0];
 assign ar_fuel_go = command[1];
 assign ar_explode_go = command[2];

 assign dac_ready = (left_chan_ready && right_chan_ready);
 assign sample_data_l = audio_out<<5;
 assign sample_data_r = audio_out<<5;

endmodule
```

sprites.mif

```
Depth = 8192;
Width = 6;
```

```
Address_radix=dec;

Data_radix=bin;

Content

BEGIN

        [0..8191] : 000000;

        0       :       000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

        16      :       000000 000000 000000 000000 000000 000000 000000 000011 000011 000000
000000 000000 000000 000000 000000 000000;

        32      :       000000 000000 000000 000000 000000 000000 000100 000101 000101 000100
000000 000000 000000 000000 000000 000000;

        48      :       000000 000000 000000 000000 000000 000000 000110 000011 000011 000110
000000 000000 000000 000000 000000 000000;

        64      :       000000 000000 000000 000000 000000 000100 000110 000011 000011 000110
000100 000000 000000 000000 000000 000000;

        80      :       000000 000000 000000 000000 000100 000111 001000 001001 001001 001000
000111 000100 000000 000000 000000 000000;

        96      :       000000 000000 000000 000100 000111 000111 001000 001001 001001 001000
000111 000111 000100 000000 000000 000000;

        112     :       000000 001010 001011 001100 001100 001000 001000 000011 000011 001000
001000 001100 001100 001011 001010 000000;

        128     :       001010 001011 001101 001110 001110 001000 001111 010000 010000 001111
001000 001110 001110 001101 001011 001010;

        144     :       001010 001011 001011 001011 001011 001000 010000 001111 001111 010000
001000 001011 001011 001011 001011 001010;

        160     :       001010 010001 010010 010010 010011 010011 010100 010100 010100 010100
010011 010011 010010 010010 010001 001010;

        176     :       010101 010110 000000 000000 010111 010111 000000 000100 000100 000000
010111 010111 000000 000000 010110 010101;

        192     :       010101 011000 000000 000000 011001 011001 000000 000000 000000 000000
011001 011001 000000 000000 000100 010101;

        208     :       000100 000000 000000 000000 011010 011010 000000 000000 000000 000000
011010 011010 000000 000000 000000 000100;
```

```
	224	:	000000 000000 000000 000000 011011 011011 000000 000000 000000 000000 011011 011011 000000 000000 000000 000000;

	240	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000;

	256	:	000000 000000 000000 000000 000000 000000 000000 001011 001100 001101 000000 000000 000000 000000 000000 000000;

	272	:	000000 000000 000000 000000 000000 000000 000000 001110 001111 010000 000000 000000 000000 000000 000000 000000;

	288	:	000000 000000 000000 000000 000000 000000 000000 010001 010010 000100 000000 000000 000000 000000 000000 000000;

	304	:	000000 000000 000000 000000 000000 000000 000000 000001 000001 000011 000000 000000 000000 000000 000000 000000;

	320	:	000000 000000 000000 000000 000000 000000 000000 000001 001001 000011 000000 000000 000000 000000 000000 000000;

	336	:	000000 000000 000000 000000 000000 000000 000000 000100 001001 001000 000000 000000 000000 000000 000000 000000;

	352	:	000000 000000 000000 000000 000000 000000 000000 000100 001001 001000 000000 000000 000000 000000 000000 000000;

	368	:	000000 000000 000000 000000 000000 000000 000000 000001 000111 000001 000000 000000 000000 000000 000000 000000;

	384	:	000000 000000 000000 000000 000000 000000 000000 000000 000001 000001 000000 000000 000000 000000 000000 000000;

	400	:	000000 000000 000000 000000 000000 000000 000000 000000 000001 000100 000000 000000 000000 000000 000000 000000;

	416	:	000000 000000 000000 000000 000000 000000 000000 000000 000001 000100 000000 000000 000000 000000 000000 000000;

	432	:	000000 000000 000000 000000 000000 000000 000000 000000 001000 000100 000000 000000 000000 000000 000000 000000;

	448	:	000000 000000 000000 000000 000000 000000 000000 000000 000001 000000 000000 000000 000000 000000 000000 000000;

	464	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000;

	480	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000;
	496	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	512	:	000000 010011 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 010011 000000;

	528	:	000000 010100 000000 000000 010101 010011 010011 010011 010011 010011
010011 010100 000000 000000 010100 000000;

	544	:	000000 010100 000000 010101 010101 010011 010110 010110 010110 010110
010011 010100 010100 000000 010100 000000;

	560	:	000000 010100 010101 010101 010101 010011 010111 010111 010111 010111
010011 010100 010100 010100 010100 000000;

	576	:	000000 010101 010101 010101 010101 010011 010011 010011 010011 010011
010011 010100 010100 010100 010100 000000;

	592	:	000000 010101 010101 011000 010101 010011 010101 010101 010101 010101
010011 010100 011001 010100 010100 000000;

	608	:	000000 010101 011000 011000 010101 010011 010011 010101 010101 010011
010011 010100 011001 011001 010100 000000;

	624	:	000000 010101 011000 011000 010101 010011 010011 010011 010011 010011
010011 010100 011001 011001 010100 000000;

	640	:	000000 010101 010101 011000 010101 010100 010100 010100 010100 010100
010100 010100 011001 010100 010100 000000;

	656	:	000000 000000 010101 010101 010101 000000 010100 010011 010011 010100
000000 010100 010100 010100 000000 000000;

	672	:	000000 000000 010101 010101 010101 000000 000000 010100 010100 000000
000000 010100 010100 010100 000000 000000;

	688	:	000000 000000 000000 010101 010101 000000 000000 000000 000000 000000
000000 010100 010100 000000 000000 000000;

	704	:	000000 000000 000000 010101 010101 000000 000000 000000 000000 000000
000000 010100 010100 000000 000000 000000;

	720	:	000000 000000 000000 000000 010101 000000 000000 000000 000000 000000
000000 010100 000000 000000 000000 000000;

	736	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	752	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;
```

```
        768    :        000000 000000 000000 000000 010101 010101 010101 010101 010101 000000
000000 000000 000000 000000 000000 000000;

        784    :        000000 000000 000000 010101 010101 010011 010101 010011 010101 010101
010101 010101 000000 000000 000000 000000;

        800    :        000000 000000 010101 010101 010101 010011 010101 010011 010101 011000
011000 011000 010101 010101 000000 000000;

        816    :        000000 000000 010101 010101 010101 010101 010101 010101 010101 010101
010101 010101 010101 010101 010101 000000;

        832    :        000000 000000 010011 010011 010011 010011 010011 010011 000000 000000
000000 000000 000000 000000 000000 000000;

        848    :        000000 010011 010011 010100 010011 010111 010110 010011 010100 010100
010011 000000 000000 000000 000000 000000;

        864    :        000000 010101 010011 010011 010011 010111 010110 010011 000000 000000
000000 000000 000000 000000 000000 000000;

        880    :        000000 010101 010011 010011 010011 010111 010110 010011 000000 000000
000000 000000 000000 000000 000000 000000;

        896    :        000000 010011 010011 010100 010011 010111 010110 010011 010100 010100
010011 000000 000000 000000 000000 000000;

        912    :        000000 000000 010011 010011 010011 010011 010011 010011 000000 000000
000000 000000 000000 000000 000000 000000;

        928    :        000000 000000 010100 010100 010100 010100 010100 010100 010100 010100
010100 010100 010100 010100 010100 000000;

        944    :        000000 000000 010100 010100 010100 010011 010100 010011 010100 011001
011001 011001 010100 010100 000000 000000;

        960    :        000000 000000 000000 010100 010100 010011 010100 010011 010100 010100
010100 010100 000000 000000 000000 000000;

        976    :        000000 000000 000000 000000 010100 010100 010100 010100 010100 000000
000000 000000 000000 000000 000000 000000;

        992    :        000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

        1008   :        000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

        1024   :        000000 000000 000000 000000 000000 000000 000000 010100 010100 010100
010100 010100 000000 000000 000000 000000;
```

```
	1040	:	000000 000000 000000 000000 010100 010100 010100 010100 010011 010100
010011 010100 010100 000000 000000 000000;

	1056	:	000000 000000 010100 010100 011001 011001 011001 010100 010011 010100
010011 010100 010100 010100 000000 000000;

	1072	:	000000 010100 010100 010100 010100 010100 010100 010100 010100 010100
010100 010100 010100 010100 000000 000000;

	1088	:	000000 000000 000000 000000 000000 000000 000000 000000 010011 010011
010011 010011 010011 010011 000000 000000;

	1104	:	000000 000000 000000 000000 000000 010011 010100 010100 010011 010110
010111 010011 010100 010011 010011 000000;

	1120	:	000000 000000 000000 000000 000000 000000 000000 000000 010011 010110
010111 010011 010011 010011 010101 000000;

	1136	:	000000 000000 000000 000000 000000 000000 000000 000000 010011 010110
010111 010011 010011 010011 010101 000000;

	1152	:	000000 000000 000000 000000 000000 010011 010100 010100 010011 010110
010111 010011 010100 010011 010011 000000;

	1168	:	000000 000000 000000 000000 000000 000000 000000 000000 010011 010011
010011 010011 010011 010011 000000 000000;

	1184	:	000000 010101 010101 010101 010101 010101 010101 010101 010101 010101
010101 010101 010101 010101 000000 000000;

	1200	:	000000 000000 010101 010101 011000 011000 011000 010101 010011 010101
010011 010101 010101 010101 000000 000000;

	1216	:	000000 000000 000000 000000 010101 010101 010101 010101 010011 010101
010011 010101 010101 000000 000000 000000;

	1232	:	000000 000000 000000 000000 000000 000000 000000 010101 010101 010101
010101 010101 000000 000000 000000 000000;

	1248	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	1264	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	1280	:	000000 000000 000000 000000 000000 010100 010100 010100 010100 010100
010100 000000 000000 000000 000000 000000;

	1296	:	000000 000000 000000 010100 010011 010011 011010 011010 011010 011010
010011 010011 010100 000000 000000 000000;
```

```
	1312	:	000000 000000 010100 010011 010011 011010 011010 011010 011010 011010
011010 010011 010011 010100 000000 000000;

	1328	:	000000 000000 010100 010100 010100 010100 010100 010100 010100 010100
010100 010100 010100 010100 000000 000000;

	1344	:	000000 000000 000000 000000 000000 000000 010100 010011 010011 010100
000000 000000 000000 000000 000000 000000;

	1360	:	000000 000000 010011 011001 010011 010011 010100 011010 011010 010100
010011 010011 011001 010011 000000 000000;

	1376	:	000000 010011 011010 011000 011010 011010 010100 010011 010011 010100
011010 011010 011000 011010 010011 000000;

	1392	:	000000 010011 011010 011000 011010 011010 010100 010011 010011 010100
011010 011010 011000 011010 010011 000000;

	1408	:	000000 000000 010011 011001 010011 010011 010100 011010 011010 010100
010011 010011 011001 010011 000000 000000;

	1424	:	000000 000000 000000 000000 000000 000000 010100 010011 010011 010100
000000 000000 000000 000000 000000 000000;

	1440	:	000000 000000 010100 010100 010100 010100 010100 010100 010100 010100
010100 010100 010100 010100 000000 000000;

	1456	:	000000 000000 010100 010011 010011 011010 011010 011010 011010 011010
011010 010011 010011 010100 000000 000000;

	1472	:	000000 000000 000000 010100 010011 010011 011010 011010 011010 011010
010011 010011 010100 000000 000000 000000;

	1488	:	000000 000000 000000 000000 000000 010100 010100 010100 010100 010100
010100 000000 000000 000000 000000 000000;

	1504	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	1520	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	1536	:	000000 000000 000000 000000 011011 011100 011101 011110 011111 100000
011011 000000 000000 000000 000000 000000;

	1552	:	000000 000000 000000 011011 100001 100010 000100 010111 010110 000100
000001 000000 000000 000000 000000 000000;

	1568	:	000000 000000 000000 100010 011111 000101 000100 001000 011000 001000
001111 011000 100010 000000 000000 000000;
```

```
	1584	:		000000 000000 011011 011111 001000 011110 010010 100010 011011 100010
011001 010111 011000 000000 000000 000000;

	1600	:		000000 000000 011011 010110 011000 100010 011000 100010 000001 011001
010111 010111 011101 000000 000000 010111;

	1616	:		000000 000000 000001 011000 001000 100010 011000 000001 100010 010111
011100 011000 010110 000000 000000 010111;

	1632	:		000000 000000 000000 100010 001111 001000 011101 011000 011000 001000
000100 010110 000000 000000 000000 010111;

	1648	:		000000 000000 000000 100010 010100 011100 011100 011110 001000 010111
010111 100010 100010 000000 000000 000000;

	1664	:		000000 000000 000000 100010 000001 011000 001000 000100 001000 010111
100000 010001 100010 000000 000000 000000;

	1680	:		000000 000000 000000 100010 000100 001000 011000 010111 001000 000100
011000 100010 100010 000000 000000 000000;

	1696	:		000000 000000 000000 000000 000001 100010 001000 001000 000100 011001
001000 001111 000001 000000 000000 000000;

	1712	:		000000 000000 000000 000000 000000 100010 010010 011010 000100 001000
000011 010111 010110 000000 000000 000000;

	1728	:		000000 000000 000000 000000 000000 000000 011011 010111 000001 000100
000000 000000 000000 000000 000000 000000;

	1744	:		000000 000000 000000 000000 000000 000000 000000 000001 000000 000000
000000 000000 000000 000000 000000 000000;

	1760	:		000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	1776	:		000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	1792	:		000000 000000 000000 000000 100011 100100 100100 100101 100011 100011
100011 100011 000000 000000 000000 000000;

	1808	:		000000 000000 000000 100101 100110 100110 100110 100111 100110 100111
100100 100100 100100 101000 000000 000000;

	1824	:		000000 000000 101000 100100 100110 100110 101001 101010 101001 101010
101010 100000 100000 100100 101000 000000;

	1840	:		000000 000000 100100 100110 101001 001001 000010 101001 100110 101001
101010 101010 100000 100100 100100 000000;
```

```
        1856    :       000000 100011 100100 100110 001001 001001 000010 100110 100110 101001
101010 001001 101000 101001 100110 000000;

        1872    :       000000 100100 100100 101010 101010 101010 001001 101001 101001 000010
100110 101010 100111 101010 100111 000000;

        1888    :       000000 100100 001011 000101 101010 101010 001001 100110 001001 101001
000010 001001 100110 000101 100110 000000;

        1904    :       000000 101000 100110 000101 101010 101010 000010 000010 100110 001001
001001 000101 100110 100110 100100 000000;

        1920    :       000000 100111 001011 001011 101010 101010 100110 001001 101001 001001
100110 101010 101010 100000 100011 000000;

        1936    :       000000 000000 100000 100110 000101 101010 101010 000010 100110 100110
101001 101010 001011 100001 101000 000000;

        1952    :       000000 000000 100100 000101 100110 000101 101010 001001 101010 101010
101010 100110 101000 001001 000000 000000;

        1968    :       000000 000000 100011 100100 100110 100100 000101 101010 000101 001011
100110 100110 100110 000101 000000 000000;

        1984    :       000000 000000 000000 100101 100110 101010 100110 100100 100100 100111
001001 000101 100100 010111 000000 000000;

        2000    :       000000 000000 000000 000000 101000 100011 100011 100011 100100 100100
100111 100100 100101 000000 000000 000000;

        2016    :       000000 000000 000000 000000 000000 000000 000000 000000 000000 100011
100011 101000 000000 000000 000000 000000;

        2032    :       000000 000000 000000 000000 000000 000000 000000 000000 100010 000001
000000 000000 000000 000000 000000 000000;

        2048    :       000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

        2064    :       000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

        2080    :       000000 000000 101011 101011 101011 101011 101011 000000 000000 101011
101011 101011 101011 101011 000000 000000;

        2096    :       000000 000000 101011 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 101011 000000 000000;

        2112    :       000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;
```

```
	2128	:	000000 000000 101011 101011 101100 101100 000010 000010 000010 000010
101101 101101 101011 101011 000000 000000;

	2144	:	000000 000000 101011 101011 101100 101100 101100 101100 101101 101101
101101 101101 101011 101011 000000 000000;

	2160	:	000000 000000 101011 101011 101100 101100 101100 101100 101101 101101
101101 101101 101011 101011 000000 000000;

	2176	:	000000 000000 101011 101011 101100 101100 101100 101100 101101 101101
101101 101101 101011 101011 000000 000000;

	2192	:	000000 000000 101011 101011 101011 101100 101100 101100 101101 101101
101101 101011 101011 101011 000000 000000;

	2208	:	000000 000000 000000 101011 101011 101011 101100 101100 101101 101101
101011 101011 101011 000000 000000 000000;

	2224	:	000000 000000 000000 000000 101011 101011 101011 101100 101101 101011
101011 101011 000000 000000 000000 000000;

	2240	:	000000 000000 000000 000000 000000 101011 101011 101011 101011 101011
101011 000000 000000 000000 000000 000000;

	2256	:	000000 000000 000000 000000 000000 000000 101011 101011 101011 101011
000000 000000 000000 000000 000000 000000;

	2272	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	2288	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	2304	:	000000 000000 000000 000000 000000 010100 010100 010100 010100 010100
010100 000000 000000 000000 000000 000000;

	2320	:	000000 000000 000000 000000 010100 010100 010100 010100 010100 010100
010100 010100 000000 000000 000000 000000;

	2336	:	000000 000000 000000 010100 010011 010011 010100 010100 010100 010100
010100 010100 010100 000000 000000 000000;

	2352	:	000000 000000 010100 010011 010011 010100 101110 101110 101110 101110
010100 010100 010100 010100 000000 000000;

	2368	:	000000 000000 010100 010011 010100 101110 101111 101111 101111 101110
101110 010100 010100 010100 000000 000000;

	2384	:	000000 000000 010100 010011 010100 101111 101111 101111 101111 101111
101110 010100 010100 010100 000000 000000;
```

```
	2400	:	000000 000000 010100 010011 010100 101111 101111 101111 101111 101111 101110 010100 010100 010100 000000 000000;
	2416	:	000000 000000 010100 010011 010100 101111 011010 101111 101111 101111 101110 010100 010011 010100 000000 000000;
	2432	:	000000 000000 010100 010011 010011 010100 101111 101111 101111 101110 010100 010011 010011 010100 000000 000000;
	2448	:	000000 000000 000000 010100 010011 010011 010100 010100 010100 010100 010011 010011 010100 000000 000000 000000;
	2464	:	000000 000000 000000 000000 010100 010011 010011 010011 010011 010011 010011 010100 000000 000000 000000 000000;
	2480	:	000000 000000 000000 000000 000000 010100 010100 010100 010100 010100 010100 000000 000000 000000 000000 000000;
	2496	:	000000 000000 000000 000000 000000 000000 000000 011010 010011 000000 000000 000000 000000 000000 000000 000000;
	2512	:	000000 000000 000000 000000 000000 000000 000000 011010 010011 000000 000000 000000 000000 000000 000000 000000;
	2528	:	000000 000000 000000 000000 000000 000000 000000 010100 010100 000000 000000 000000 000000 000000 000000 000000;
	2544	:	000000 000000 000000 000000 000000 000000 110000 110001 000000 000000 000000 000000 000000 000000 000000 000000;
	2560	:	000000 000000 000000 000000 000000 101011 101011 101011 101011 101011 101011 000000 000000 000000 000000 000000;
	2576	:	000000 000000 000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 000000 000000 000000 000000;
	2592	:	000000 000000 000000 101011 101011 101011 110010 110010 110010 110010 101011 101011 101011 000000 000000 000000;
	2608	:	000000 000000 000000 101011 101011 110011 110010 110010 110010 110010 110011 101011 101011 000000 000000 000000;
	2624	:	000000 000000 000000 101011 101011 110011 110010 110010 110010 110010 110011 101011 101011 000000 000000 000000;
	2640	:	000000 000000 000000 101011 101011 110011 110011 110011 110011 110011 110011 101011 101011 000000 000000 000000;
	2656	:	000000 000000 000000 101011 101011 110011 110011 110100 110100 110011 110011 101011 101011 000000 000000 000000;
	2672	:	000000 000000 000000 101011 101011 110011 110011 110100 110100 110011 110011 101011 101011 000000 000000 000000;

	2688	:	000000 000000 000000 101011 101011 110011 110011 110100 110100 110011 110011 101011 101011 000000 000000 000000;

	2704	:	000000 000000 000000 101011 101011 110011 110011 110011 110011 110011 110011 101011 101011 000000 000000 000000;

	2720	:	000000 000000 000000 101011 101011 101011 101011 110101 110101 101011 101011 101011 101011 000000 000000 000000;

	2736	:	000000 000000 000000 101011 101011 101011 101011 110010 110010 101011 101011 101011 101011 000000 000000 000000;

	2752	:	000000 000000 000000 000000 000000 101011 101011 110101 110101 101011 101011 000000 000000 000000 000000 000000;

	2768	:	000000 000000 000000 000000 000000 101011 101011 110101 110101 101011 101011 000000 000000 000000 000000 000000;

	2784	:	000000 000000 000000 000000 000000 101011 101011 101011 101011 101011 101011 000000 000000 000000 000000 000000;

	2800	:	000000 000000 000000 000000 000000 101011 101011 101011 101011 101011 101011 000000 000000 000000 000000 000000;

	2816	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000;

	2832	:	000000 000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 000000 000000 000000;

	2848	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 000000 000000;

	2864	:	000000 000000 101011 101011 110110 110110 000010 000010 000010 000010 110110 110110 101011 101011 000000 000000;

	2880	:	000000 000000 101011 101011 110110 110110 000010 000010 000010 000010 110110 110110 101011 101011 000000 000000;

	2896	:	000000 000000 101011 101011 101100 101100 101101 101101 101101 101101 101100 101100 101011 101011 000000 000000;

	2912	:	000000 000000 101011 101011 101100 101100 000010 000010 000010 000010 101100 101100 101011 101011 000000 000000;

	2928	:	000000 000000 101011 101011 101100 101100 000010 000010 101101 000010 101100 101100 101011 101011 000000 000000;
	2944	:	000000 000000 101011 101011 101100 101100 000010 000010 000010 000010
101100 101100 101011 101011 000000 000000;

	2960	:	000000 000000 101011 101011 101100 101100 000010 000010 101101 101101
101100 101100 101011 101011 000000 000000;

	2976	:	000000 000000 101011 101011 101100 101100 101101 101101 101101 101101
101100 101100 101011 101011 000000 000000;

	2992	:	000000 000000 101011 101011 110110 110110 000010 000010 000010 000010
110110 110110 101011 101011 000000 000000;

	3008	:	000000 000000 101011 101011 110110 110110 000010 000010 000010 000010
110110 110110 101011 101011 000000 000000;

	3024	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 101011 000000 000000;

	3040	:	000000 000000 000000 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 000000 000000 000000;

	3056	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	3072	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	3088	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	3104	:	000000 101011 101011 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 101011 101011 000000;

	3120	:	101011 101011 101011 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 101011 101011 101011;

	3136	:	101011 101011 110110 110110 110100 110100 110100 110100 110100 110100
110100 110100 110110 110110 101011 101011;

	3152	:	101011 101011 110110 110110 110100 110100 110100 000010 000010 110100
110100 110100 110110 110110 101011 101011;

	3168	:	101011 101011 000010 000010 110011 110011 110011 000010 000010 110011
110011 110011 000010 000010 101011 101011;

	3184	:	101011 101011 000010 000010 110011 000010 000010 000010 000010 000010
000010 110011 000010 000010 101011 101011;

	3200	:	101011 101011 000010 000010 110011 000010 000010 000010 000010 000010
000010 110011 000010 000010 101011 101011;
```

```
        3216    :       101011 101011 000010 000010 110011 110011 110011 000010 000010 110011
110011 110011 000010 000010 101011 101011;

        3232    :       101011 101011 110110 110110 110100 110100 110100 000010 000010 110100
110100 110100 110110 110110 101011 101011;

        3248    :       101011 101011 110110 110110 110100 110100 110100 110100 110100 110100
110100 110100 110110 110110 101011 101011;

        3264    :       101011 101011 101011 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 101011 101011 101011;

        3280    :       000000 101011 101011 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 101011 101011 000000;

        3296    :       000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

        3312    :       000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

        3328    :       000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

        3344    :       000000 000000 000000 000000 101011 101011 101011 101011 101011 101011
101011 101011 000000 000000 000000 000000;

        3360    :       000000 000000 000000 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 000000 000000 000000;

        3376    :       000000 000000 101011 101011 101011 000010 110110 000010 000010 000010
000010 101011 101011 101011 000000 000000;

        3392    :       000000 000000 101011 101011 000010 000010 110110 000010 000010 000010
000010 000010 101011 101011 000000 000000;

        3408    :       000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;

        3424    :       000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;

        3440    :       000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;

        3456    :       000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;

        3472    :       000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;
```

```
	3488	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;

	3504	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 110110
000010 000010 101011 101011 000000 000000;

	3520	:	000000 000000 101011 101011 101011 000010 000010 000010 000010 110110
000010 101011 101011 101011 000000 000000;

	3536	:	000000 000000 000000 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 000000 000000 000000;

	3552	:	000000 000000 000000 000000 101011 101011 101011 101011 101011 101011
101011 101011 000000 000000 000000 000000;

	3568	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	3584	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	3600	:	000000 000000 000000 101011 101011 101011 101011 101011 101011 101011
101011 000000 000000 000000 000000 000000;

	3616	:	000000 000000 000000 101011 101011 101011 101011 101011 101011 101011
101011 000000 000000 000000 000000 000000;

	3632	:	000000 000000 000000 101011 101011 000010 110110 000010 000010 101011
101011 000000 000000 000000 000000 000000;

	3648	:	000000 000000 000000 101011 101011 000010 110110 000010 000010 101011
101011 000000 000000 000000 000000 000000;

	3664	:	000000 000000 000000 101011 101011 101011 101011 000010 000010 101011
101011 000000 000000 000000 000000 000000;

	3680	:	000000 000000 000000 101011 101011 101011 101011 000010 000010 101011
101011 000000 000000 000000 000000 000000;

	3696	:	000000 000000 000000 000000 000000 101011 101011 000010 000010 101011
101011 000000 000000 000000 000000 000000;

	3712	:	000000 000000 000000 000000 000000 101011 101011 000010 000010 101011
101011 000000 000000 000000 000000 000000;

	3728	:	000000 000000 000000 000000 000000 101011 101011 000010 000010 101011
101011 000000 000000 000000 000000 000000;

	3744	:	000000 000000 000000 000000 000000 101011 101011 000010 000010 101011
101011 000000 000000 000000 000000 000000;
```

```
	3760	:	000000 000000 000000 000000 000000 101011 101011 000010 000010 101011
101011 000000 000000 000000 000000 000000;

	3776	:	000000 000000 000000 000000 000000 101011 101011 000010 000010 101011
101011 000000 000000 000000 000000 000000;

	3792	:	000000 000000 000000 000000 000000 101011 101011 101011 101011 101011
101011 000000 000000 000000 000000 000000;

	3808	:	000000 000000 000000 000000 000000 101011 101011 101011 101011 101011
101011 000000 000000 000000 000000 000000;

	3824	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	3840	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	3856	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011
101011 101011 000000 000000 000000 000000;

	3872	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 000000 000000 000000;

	3888	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010
000010 101011 101011 101011 000000 000000;

	3904	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010
000010 000010 101011 101011 000000 000000;

	3920	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;

	3936	:	000000 000000 000000 101011 101011 101011 101011 101011 101011 101011
110110 110110 101011 101011 000000 000000;

	3952	:	000000 000000 101011 101011 101011 000010 000010 000010 000010 000010
000010 000010 101011 101011 000000 000000;

	3968	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010
000010 101011 101011 101011 000000 000000;

	3984	:	000000 000000 101011 101011 110110 110110 101011 101011 101011 101011
101011 101011 101011 000000 000000 000000;

	4000	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
101011 101011 101011 101011 000000 000000;

	4016	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010
000010 000010 101011 101011 000000 000000;
```

```
	4032	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010 000010 000010 101011 101011 000000 000000;
	4048	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 000000 000000;
	4064	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 000000 000000;
	4080	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000;
	4096	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000;
	4112	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 000000 000000 000000 000000;
	4128	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 000000 000000 000000;
	4144	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010 000010 101011 101011 101011 000000 000000;
	4160	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010 000010 000010 101011 101011 000000 000000;
	4176	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 000010 000010 101011 101011 000000 000000;
	4192	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 000010 000010 101011 101011 000000 000000;
	4208	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 110110 000010 000010 101011 101011 000000 000000;
	4224	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 110110 000010 000010 101011 101011 000000 000000;
	4240	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 000010 000010 101011 101011 000000 000000;
	4256	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 000010 000010 101011 101011 000000 000000;
	4272	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010 000010 000010 101011 101011 000000 000000;
	4288	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010 000010 101011 101011 101011 000000 000000;
	4304	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 000000 000000 000000;

	4320	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 000000 000000 000000 000000;

	4336	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000;

	4352	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000;

	4368	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 000000 000000;

	4384	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 000000 000000;

	4400	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011 000010 000010 101011 101011 000000 000000;

	4416	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011 000010 000010 101011 101011 000000 000000;

	4432	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011 000010 000010 101011 101011 000000 000000;

	4448	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011 000010 000010 101011 101011 000000 000000;

	4464	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 110110 000010 000010 101011 101011 000000 000000;

	4480	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 110110 000010 000010 101011 101011 000000 000000;

	4496	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 000010 000010 101011 101011 000000 000000;

	4512	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 000010 000010 101011 101011 000000 000000;

	4528	:	000000 000000 000000 000000 000000 000000 000000 000000 101011 101011 000010 000010 101011 101011 000000 000000;

	4544	:	000000 000000 000000 000000 000000 000000 000000 000000 101011 101011 000010 000010 101011 101011 000000 000000;

	4560	:	000000 000000 000000 000000 000000 000000 000000 000000 101011 101011 101011 101011 101011 101011 000000 000000;
	4576	:	000000 000000 000000 000000 000000 000000 000000 000000 101011 101011
101011 101011 101011 101011 000000 000000;

	4592	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	4608	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	4624	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 101011 000000 000000;

	4640	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 101011 000000 000000;

	4656	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010
000010 000010 101011 101011 000000 000000;

	4672	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010
000010 000010 101011 101011 000000 000000;

	4688	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
101011 101011 101011 101011 000000 000000;

	4704	:	000000 000000 101011 101011 110110 110110 101011 101011 101011 101011
101011 101011 101011 000000 000000 000000;

	4720	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010
000010 101011 101011 101011 000000 000000;

	4736	:	000000 000000 101011 101011 101011 000010 000010 000010 000010 000010
000010 000010 101011 101011 000000 000000;

	4752	:	000000 000000 000000 101011 101011 101011 101011 101011 101011 101011
110110 110110 101011 101011 000000 000000;

	4768	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;

	4784	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010
000010 000010 101011 101011 000000 000000;

	4800	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010
000010 101011 101011 101011 000000 000000;

	4816	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 000000 000000 000000;

	4832	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011
101011 101011 000000 000000 000000 000000;
```

```
	4848	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	4864	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	4880	:	000000 000000 000000 000000 000000 101011 101011 101011 101011 101011
101011 000000 000000 000000 000000 000000;

	4896	:	000000 000000 000000 000000 101011 101011 101011 101011 101011 101011
101011 000000 000000 000000 000000 000000;

	4912	:	000000 000000 000000 101011 101011 101011 000010 000010 000010 101011
101011 000000 000000 000000 000000 000000;

	4928	:	000000 000000 101011 101011 101011 000010 000010 000010 101011 101011
101011 000000 000000 000000 000000 000000;

	4944	:	000000 000000 101011 101011 000010 000010 000010 101011 101011 101011
101011 101011 000000 000000 000000 000000;

	4960	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
101011 101011 101011 000000 000000 000000;

	4976	:	000000 000000 101011 101011 000010 000010 110110 000010 000010 000010
000010 101011 101011 101011 000000 000000;

	4992	:	000000 000000 101011 101011 000010 000010 110110 000010 000010 000010
000010 000010 101011 101011 000000 000000;

	5008	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;

	5024	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;

	5040	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010
000010 000010 101011 101011 000000 000000;

	5056	:	000000 000000 101011 101011 101011 000010 000010 000010 000010 000010
000010 101011 101011 101011 000000 000000;

	5072	:	000000 000000 000000 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 000000 000000 000000;

	5088	:	000000 000000 000000 000000 101011 101011 101011 101011 101011 101011
101011 101011 000000 000000 000000 000000;

	5104	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;
```

```
        5120    :       000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

        5136    :       000000 000000 101011 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 000000 000000 000000;

        5152    :       000000 000000 101011 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 101011 000000 000000;

        5168    :       000000 000000 101011 101011 000010 000010 000010 000010 000010 110110
000010 000010 101011 101011 000000 000000;

        5184    :       000000 000000 101011 101011 000010 000010 000010 000010 000010 110110
000010 000010 101011 101011 000000 000000;

        5200    :       000000 000000 101011 101011 101011 101011 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;

        5216    :       000000 000000 101011 101011 101011 101011 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;

        5232    :       000000 000000 000000 000000 000000 000000 101011 101011 101011 000010
000010 000010 101011 101011 000000 000000;

        5248    :       000000 000000 000000 000000 000000 101011 101011 101011 000010 000010
000010 101011 101011 101011 000000 000000;

        5264    :       000000 000000 000000 000000 000000 101011 101011 000010 000010 000010
101011 101011 101011 000000 000000 000000;

        5280    :       000000 000000 000000 000000 000000 101011 101011 000010 000010 101011
101011 101011 000000 000000 000000 000000;

        5296    :       000000 000000 000000 000000 000000 101011 101011 000010 000010 101011
101011 000000 000000 000000 000000 000000;

        5312    :       000000 000000 000000 000000 000000 101011 101011 000010 000010 101011
101011 000000 000000 000000 000000 000000;

        5328    :       000000 000000 000000 000000 000000 101011 101011 101011 101011 101011
101011 000000 000000 000000 000000 000000;

        5344    :       000000 000000 000000 000000 000000 101011 101011 101011 101011 101011
101011 000000 000000 000000 000000 000000;

        5360    :       000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

        5376    :       000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;
```

```
	5392	:	000000 000000 000000 000000 101011 101011 101011 101011 101011 101011
101011 101011 000000 000000 000000 000000;
	5408	:	000000 000000 000000 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 000000 000000 000000;
	5424	:	000000 000000 101011 101011 101011 000010 000010 000010 000010 000010
000010 101011 101011 101011 000000 000000;
	5440	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010
000010 000010 101011 101011 000000 000000;
	5456	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;
	5472	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;
	5488	:	000000 000000 101011 101011 101011 000010 110110 000010 000010 110110
000010 101011 101011 101011 000000 000000;
	5504	:	000000 000000 101011 101011 101011 000010 110110 000010 000010 110110
000010 101011 101011 101011 000000 000000;
	5520	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;
	5536	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;
	5552	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010
000010 000010 101011 101011 000000 000000;
	5568	:	000000 000000 101011 101011 101011 000010 000010 000010 000010 000010
000010 101011 101011 101011 000000 000000;
	5584	:	000000 000000 000000 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 000000 000000 000000;
	5600	:	000000 000000 000000 000000 101011 101011 101011 101011 101011 101011
101011 101011 000000 000000 000000 000000;
	5616	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;
	5632	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;
	5648	:	000000 000000 000000 000000 101011 101011 101011 101011 101011 101011
101011 101011 000000 000000 000000 000000;
```

```
	5664	:	000000 000000 000000 101011 101011 101011 101011 101011 101011 101011
101011 101011 101011 000000 000000 000000;

	5680	:	000000 000000 101011 101011 101011 000010 000010 000010 000010 000010
000010 101011 101011 101011 000000 000000;

	5696	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 000010
000010 000010 101011 101011 000000 000000;

	5712	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;

	5728	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;

	5744	:	000000 000000 101011 101011 000010 000010 000010 000010 000010 110110
000010 000010 101011 101011 000000 000000;

	5760	:	000000 000000 101011 101011 101011 000010 000010 000010 000010 110110
000010 000010 101011 101011 000000 000000;

	5776	:	000000 000000 000000 101011 101011 101011 101011 101011 101011 101011
000010 000010 101011 101011 000000 000000;

	5792	:	000000 000000 000000 000000 101011 101011 101011 101011 101011 000010
000010 000010 101011 101011 000000 000000;

	5808	:	000000 000000 000000 000000 000000 101011 101011 101011 000010 000010
000010 101011 101011 101011 000000 000000;

	5824	:	000000 000000 000000 000000 000000 101011 101011 000010 000010 000010
101011 101011 101011 000000 000000 000000;

	5840	:	000000 000000 000000 000000 000000 101011 101011 101011 101011 101011
101011 101011 000000 000000 000000 000000;

	5856	:	000000 000000 000000 000000 000000 101011 101011 101011 101011 101011
101011 000000 000000 000000 000000 000000;

	5872	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	5888	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	5904	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000;

	5920	:	000000 000000 000000 101011 101011 101011 101011 000000 000000 101011
101011 101011 101011 000000 000000 000000;
```

```
	5936	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 000000 000000;
	5952	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011 000010 000010 101011 101011 000000 000000;
	5968	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 000010 000010 000010 101011 101011 000000 000000;
	5984	:	000000 000000 101011 101011 101011 101011 101011 101011 000010 000010 000010 101011 101011 101011 000000 000000;
	6000	:	000000 000000 000000 101011 101011 101011 101011 000010 000010 000010 101011 101011 101011 000000 000000 000000;
	6016	:	000000 000000 000000 101011 101011 101011 000010 000010 000010 101011 101011 101011 101011 000000 000000 000000;
	6032	:	000000 000000 101011 101011 101011 000010 000010 000010 101011 101011 101011 101011 101011 101011 000000 000000;
	6048	:	000000 000000 101011 101011 000010 000010 000010 101011 101011 101011 000010 000010 101011 101011 000000 000000;
	6064	:	000000 000000 101011 101011 000010 000010 101011 101011 101011 101011 000010 000010 101011 101011 000000 000000;
	6080	:	000000 000000 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 101011 000000 000000;
	6096	:	000000 000000 000000 101011 101011 101011 101011 000000 000000 101011 101011 101011 101011 000000 000000 000000;
	6112	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000;
	6128	:	000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000;
END;
```

audioROM_fire.v

```
// megafunction wizard: %ROM: 1-PORT%
// GENERATION: STANDARD

// VERSION: WM1.0

// MODULE: altsyncram


// ============================================================

// File Name: audioROM_fire.v

// Megafunction Name(s):

// 			altsyncram

//

// Simulation Library Files(s):

// 			altera_mf

// ============================================================

// ************************************************************

// THIS IS A WIZARD-GENERATED FILE. DO NOT EDIT THIS FILE!

//

// 21.1.0 Build 842 10/21/2021 SJ Lite Edition

// ************************************************************



//Copyright (C) 2021  Intel Corporation. All rights reserved.

//Your use of Intel Corporation's design tools, logic functions

//and other software and tools, and any partner logic

//functions, and any output files from any of the foregoing

//(including device programming or simulation files), and any

//associated documentation or information are expressly subject

//to the terms and conditions of the Intel Program License

//Subscription Agreement, the Intel Quartus Prime License Agreement,

//the Intel FPGA IP License Agreement, or other applicable license

//agreement, including, without limitation, that your use is for
```

```
//the sole purpose of programming logic devices manufactured by
//Intel and sold by Intel or its authorized distributors.  Please
//refer to the applicable agreement for further details, at
//https://fpgasoftware.intel.com/eula.



// synopsys translate_off
`timescale 1 ps / 1 ps
// synopsys translate_on
module audioROM_fire (
	address,
	clock,
	q);

	input	[10:0]  address;
	input	  clock;
	output	[7:0]  q;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_off
`endif
	tri1	  clock;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_on
`endif

	wire [7:0] sub_wire0;
	wire [7:0] q = sub_wire0[7:0];

	altsyncram	altsyncram_component (
                .address_a (address),
                .clock0 (clock),
                .q_a (sub_wire0),
                .aclr0 (1'b0),
                .aclr1 (1'b0),
                .address_b (1'b1),
                .addressstall_a (1'b0),
                .addressstall_b (1'b0),
                .byteena_a (1'b1),
                .byteena_b (1'b1),
                .clock1 (1'b1),
                .clocken0 (1'b1),
                .clocken1 (1'b1),
                .clocken2 (1'b1),
                .clocken3 (1'b1),
                .data_a ({8{1'b1}}),
                .data_b (1'b1),
                .eccstatus (),
                .q_b (),
                .rden_a (1'b1),
                .rden_b (1'b1),
                .wren_a (1'b0),
                .wren_b (1'b0));
    defparam
        altsyncram_component.address_aclr_a = "NONE",
        altsyncram_component.clock_enable_input_a = "BYPASS",
        altsyncram_component.clock_enable_output_a = "BYPASS",
`ifdef NO_PLI
        altsyncram_component.init_file = "rr_audio_fire.rif"
```

```
`else
		altsyncram_component.init_file = "rr_audio_fire.hex"
`endif
,
		altsyncram_component.intended_device_family = "Cyclone V",
		altsyncram_component.lpm_hint = "ENABLE_RUNTIME_MOD=NO",
		altsyncram_component.lpm_type = "altsyncram",
		altsyncram_component.numwords_a = 1600,
		altsyncram_component.operation_mode = "ROM",
		altsyncram_component.outdata_aclr_a = "NONE",
		altsyncram_component.outdata_reg_a = "UNREGISTERED",
		altsyncram_component.widthad_a = 11,
		altsyncram_component.width_a = 8,
		altsyncram_component.width_byteena_a = 1;


endmodule

// ============================================================
// CNX file retrieval info
// ============================================================
// Retrieval info: PRIVATE: ADDRESSSTALL_A NUMERIC "0"
// Retrieval info: PRIVATE: AclrAddr NUMERIC "0"
// Retrieval info: PRIVATE: AclrByte NUMERIC "0"
// Retrieval info: PRIVATE: AclrOutput NUMERIC "0"
// Retrieval info: PRIVATE: BYTE_ENABLE NUMERIC "0"
// Retrieval info: PRIVATE: BYTE_SIZE NUMERIC "8"
// Retrieval info: PRIVATE: BlankMemory NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_INPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_OUTPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: Clken NUMERIC "0"
// Retrieval info: PRIVATE: IMPLEMENT_IN_LES NUMERIC "0"
// Retrieval info: PRIVATE: INIT_FILE_LAYOUT STRING "PORT_A"
// Retrieval info: PRIVATE: INIT_TO_SIM_X NUMERIC "0"
// Retrieval info: PRIVATE: INTENDED_DEVICE_FAMILY STRING "Cyclone V"
// Retrieval info: PRIVATE: JTAG_ENABLED NUMERIC "0"
// Retrieval info: PRIVATE: JTAG_ID STRING "NONE"
// Retrieval info: PRIVATE: MAXIMUM_DEPTH NUMERIC "0"
// Retrieval info: PRIVATE: MIFfilename STRING "rr_audio_fire.hex"
// Retrieval info: PRIVATE: NUMWORDS_A NUMERIC "1600"
// Retrieval info: PRIVATE: RAM_BLOCK_TYPE NUMERIC "0"
// Retrieval info: PRIVATE: RegAddr NUMERIC "1"
// Retrieval info: PRIVATE: RegOutput NUMERIC "0"
// Retrieval info: PRIVATE: SYNTH_WRAPPER_GEN_POSTFIX STRING "0"
// Retrieval info: PRIVATE: SingleClock NUMERIC "1"
// Retrieval info: PRIVATE: UseDQRAM NUMERIC "0"
// Retrieval info: PRIVATE: WidthAddr NUMERIC "11"
// Retrieval info: PRIVATE: WidthData NUMERIC "8"
// Retrieval info: PRIVATE: rden NUMERIC "0"
// Retrieval info: LIBRARY: altera_mf altera_mf.altera_mf_components.all
// Retrieval info: CONSTANT: ADDRESS_ACLR_A STRING "NONE"
// Retrieval info: CONSTANT: CLOCK_ENABLE_INPUT_A STRING "BYPASS"
// Retrieval info: CONSTANT: CLOCK_ENABLE_OUTPUT_A STRING "BYPASS"
// Retrieval info: CONSTANT: INIT_FILE STRING "rr_audio_fire.hex"
// Retrieval info: CONSTANT: INTENDED_DEVICE_FAMILY STRING "Cyclone V"
// Retrieval info: CONSTANT: LPM_HINT STRING "ENABLE_RUNTIME_MOD=NO"
// Retrieval info: CONSTANT: LPM_TYPE STRING "altsyncram"
// Retrieval info: CONSTANT: NUMWORDS_A NUMERIC "1600"
// Retrieval info: CONSTANT: OPERATION_MODE STRING "ROM"
// Retrieval info: CONSTANT: OUTDATA_ACLR_A STRING "NONE"
// Retrieval info: CONSTANT: OUTDATA_REG_A STRING "UNREGISTERED"
// Retrieval info: CONSTANT: WIDTHAD_A NUMERIC "11"
// Retrieval info: CONSTANT: WIDTH_A NUMERIC "8"
// Retrieval info: CONSTANT: WIDTH_BYTEENA_A NUMERIC "1"
// Retrieval info: USED_PORT: address 0 0 11 0 INPUT NODEFVAL "address[10..0]"
// Retrieval info: USED_PORT: clock 0 0 0 0 INPUT VCC "clock"
// Retrieval info: USED_PORT: q 0 0 8 0 OUTPUT NODEFVAL "q[7..0]"
// Retrieval info: CONNECT: @address_a 0 0 11 0 address 0 0 11 0
// Retrieval info: CONNECT: @clock0 0 0 0 0 clock 0 0 0 0
// Retrieval info: CONNECT: q 0 0 8 0 @q_a 0 0 8 0
// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fire.v TRUE
// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fire.inc FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fire.cmp FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fire.bsf FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fire_inst.v FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fire_bb.v TRUE
// Retrieval info: LIB_FILE: altera_mf
```

audioROM_fly.v

```
// megafunction wizard: %ROM: 1-PORT%
// GENERATION: STANDARD
// VERSION: WM1.0
// MODULE: altsyncram
// ============================================================
// File Name: audioROM_fly.v
// Megafunction Name(s):
//                      altsyncram
//
// Simulation Library Files(s):
//                      altera_mf
// ============================================================
// ************************************************************
// THIS IS A WIZARD-GENERATED FILE. DO NOT EDIT THIS FILE!
//
// 21.1.0 Build 842 10/21/2021 SJ Lite Edition
// ************************************************************


//Copyright (C) 2021  Intel Corporation. All rights reserved.
//Your use of Intel Corporation's design tools, logic functions
//and other software and tools, and any partner logic
//functions, and any output files from any of the foregoing
//(including device programming or simulation files), and any
//associated documentation or information are expressly subject
//to the terms and conditions of the Intel Program License
//Subscription Agreement, the Intel Quartus Prime License Agreement,
//the Intel FPGA IP License Agreement, or other applicable license
//agreement, including, without limitation, that your use is for
//the sole purpose of programming logic devices manufactured by
//Intel and sold by Intel or its authorized distributors.  Please
//refer to the applicable agreement for further details, at
```

```
//https://fpgasoftware.intel.com/eula.



// synopsys translate_off
`timescale 1 ps / 1 ps
// synopsys translate_on
module audioROM_fly (
        address,
        clock,
        q);

        input    [14:0]  address;
        input      clock;
        output  [7:0]  q;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_off
`endif
        tri1        clock;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_on
`endif

        wire [7:0] sub_wire0;
        wire [7:0] q = sub_wire0[7:0];

        altsyncram        altsyncram_component (
                                .address_a (address),
                                .clock0 (clock),
                                .q_a (sub_wire0),
                                .aclr0 (1'b0),
                                .aclr1 (1'b0),
                                .address_b (1'b1),
                                .addressstall_a (1'b0),
                                .addressstall_b (1'b0),
                                .byteena_a (1'b1),
                                .byteena_b (1'b1),
                                .clock1 (1'b1),
                                .clocken0 (1'b1),
                                .clocken1 (1'b1),
                                .clocken2 (1'b1),
                                .clocken3 (1'b1),
                                .data_a ({8{1'b1}}),
                                .data_b (1'b1),
                                .eccstatus (),
                                .q_b (),
                                .rden_a (1'b1),
                                .rden_b (1'b1),
                                .wren_a (1'b0),
                                .wren_b (1'b0));
        defparam
                  altsyncram_component.address_aclr_a = "NONE",
                  altsyncram_component.clock_enable_input_a = "BYPASS",
                  altsyncram_component.clock_enable_output_a = "BYPASS",
`ifdef NO_PLI
                  altsyncram_component.init_file = "rr_audio_fly.rif"
`else
                  altsyncram_component.init_file = "rr_audio_fly.hex"
`endif
```

```
,
                altsyncram_component.intended_device_family = "Cyclone V",
                altsyncram_component.lpm_hint = "ENABLE_RUNTIME_MOD=NO",
                altsyncram_component.lpm_type = "altsyncram",
                altsyncram_component.numwords_a = 23214,
                altsyncram_component.operation_mode = "ROM",
                altsyncram_component.outdata_aclr_a = "NONE",
                altsyncram_component.outdata_reg_a = "UNREGISTERED",
                altsyncram_component.widthad_a = 15,
                altsyncram_component.width_a = 8,
                altsyncram_component.width_byteena_a = 1;


endmodule

// ============================================================
// CNX file retrieval info
// ============================================================
// Retrieval info: PRIVATE: ADDRESSSTALL_A NUMERIC "0"
// Retrieval info: PRIVATE: AclrAddr NUMERIC "0"
// Retrieval info: PRIVATE: AclrByte NUMERIC "0"
// Retrieval info: PRIVATE: AclrOutput NUMERIC "0"
// Retrieval info: PRIVATE: BYTE_ENABLE NUMERIC "0"
// Retrieval info: PRIVATE: BYTE_SIZE NUMERIC "8"
// Retrieval info: PRIVATE: BlankMemory NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_INPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_OUTPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: Clken NUMERIC "0"
// Retrieval info: PRIVATE: IMPLEMENT_IN_LES NUMERIC "0"
// Retrieval info: PRIVATE: INIT_FILE_LAYOUT STRING "PORT_A"

// Retrieval info: PRIVATE: INIT_TO_SIM_X NUMERIC "0"

// Retrieval info: PRIVATE: INTENDED_DEVICE_FAMILY STRING "Cyclone V"

// Retrieval info: PRIVATE: JTAG_ENABLED NUMERIC "0"

// Retrieval info: PRIVATE: JTAG_ID STRING "NONE"

// Retrieval info: PRIVATE: MAXIMUM_DEPTH NUMERIC "0"

// Retrieval info: PRIVATE: MIFfilename STRING "rr_audio_fly.hex"

// Retrieval info: PRIVATE: NUMWORDS_A NUMERIC "23214"

// Retrieval info: PRIVATE: RAM_BLOCK_TYPE NUMERIC "0"

// Retrieval info: PRIVATE: RegAddr NUMERIC "1"

// Retrieval info: PRIVATE: RegOutput NUMERIC "0"

// Retrieval info: PRIVATE: SYNTH_WRAPPER_GEN_POSTFIX STRING "0"

// Retrieval info: PRIVATE: SingleClock NUMERIC "1"

// Retrieval info: PRIVATE: UseDQRAM NUMERIC "0"

// Retrieval info: PRIVATE: WidthAddr NUMERIC "15"

// Retrieval info: PRIVATE: WidthData NUMERIC "8"

// Retrieval info: PRIVATE: rden NUMERIC "0"

// Retrieval info: LIBRARY: altera_mf altera_mf.altera_mf_components.all

// Retrieval info: CONSTANT: ADDRESS_ACLR_A STRING "NONE"

// Retrieval info: CONSTANT: CLOCK_ENABLE_INPUT_A STRING "BYPASS"

// Retrieval info: CONSTANT: CLOCK_ENABLE_OUTPUT_A STRING "BYPASS"

// Retrieval info: CONSTANT: INIT_FILE STRING "rr_audio_fly.hex"

// Retrieval info: CONSTANT: INTENDED_DEVICE_FAMILY STRING "Cyclone V"

// Retrieval info: CONSTANT: LPM_HINT STRING "ENABLE_RUNTIME_MOD=NO"

// Retrieval info: CONSTANT: LPM_TYPE STRING "altsyncram"

// Retrieval info: CONSTANT: NUMWORDS_A NUMERIC "23214"

// Retrieval info: CONSTANT: OPERATION_MODE STRING "ROM"

// Retrieval info: CONSTANT: OUTDATA_ACLR_A STRING "NONE"

// Retrieval info: CONSTANT: OUTDATA_REG_A STRING "UNREGISTERED"
// Retrieval info: CONSTANT: WIDTHAD_A NUMERIC "15"

// Retrieval info: CONSTANT: WIDTH_A NUMERIC "8"

// Retrieval info: CONSTANT: WIDTH_BYTEENA_A NUMERIC "1"

// Retrieval info: USED_PORT: address 0 0 15 0 INPUT NODEFVAL "address[14..0]"

// Retrieval info: USED_PORT: clock 0 0 0 0 INPUT VCC "clock"

// Retrieval info: USED_PORT: q 0 0 8 0 OUTPUT NODEFVAL "q[7..0]"

// Retrieval info: CONNECT: @address_a 0 0 15 0 address 0 0 15 0

// Retrieval info: CONNECT: @clock0 0 0 0 0 clock 0 0 0 0

// Retrieval info: CONNECT: q 0 0 8 0 @q_a 0 0 8 0

// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fly.v TRUE

// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fly.inc FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fly.cmp FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fly.bsf FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fly_inst.v FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fly_bb.v TRUE

// Retrieval info: LIB_FILE: altera_mf
```

shoot.mif

```
WIDTH = 16;

DEPTH = 4978;

ADDRESS_RADIX = HEX;

DATA_RADIX = DEC;

CONTENT BEGIN


0 : -1;
```

```
1 : -1;
2 : -1;
3 : -1;
4 : -1;
5 : -1;
6 : -1;
7 : -1;
8 : -1;
9 : -1;
a : -1;
b : -1;
c : -1;
d : -1;
e : -1;
f : -1;
10 : -1;
11 : -1;
12 : -1;
13 : -1;
14 : -1;
15 : -1;
16 : -1;
17 : -1;
18 : -1;
19 : -1;
1a : -1;
1b : -1;
1c : -1;
1d : -1;
```

```
1e : -1;

1f : -1;

20 : -1;

21 : -1;

22 : -1;

23 : -1;

24 : 0;

25 : -2;

26 : 0;

27 : -3;

28 : 3;

29 : -7;

2a : 10;

2b : -41;

2c : -142;

2d : -123;

2e : -133;

2f : -127;

30 : -131;

31 : -128;

32 : -130;

33 : -129;

34 : -129;

35 : -129;

36 : -129;

37 : -129;

38 : -129;

39 : -128;

3a : -129;
```

```
3b : -128;
3c : -130;
3d : -126;
3e : -141;
3f : -257;
40 : -142;
41 : -369;
42 : -513;
43 : -401;
44 : -433;
45 : -201;
46 : -252;
47 : -322;
48 : -270;
49 : -327;
4a : -235;
4b : -327;
4c : -627;
4d : -471;
4e : -562;
4f : -486;
50 : -646;
51 : -645;
52 : -422;
53 : -401;
54 : -209;
55 : -324;
56 : -289;
57 : -273;
```

```
58 : -247;
59 : -138;
5a : -129;
5b : -124;
5c : -135;
5d : -174;
5e : -344;
5f : -136;
60 : -120;
61 : -79;
62 : -36;
63 : -78;
64 : -68;
65 : 14;
66 : -39;
67 : -74;
68 : 6;
69 : 71;
6a : 46;
6b : 225;
6c : 303;
6d : 546;
6e : 398;
6f : 259;
70 : 117;
71 : -28;
72 : 11;
73 : -6;
74 : -4;
```

```
75 : 11;

76 : -78;

77 : -130;

78 : 19;

79 : -21;

7a : 34;

7b : 370;

7c : 853;

7d : 1250;

7e : 1323;

7f : 996;

80 : 513;

81 : 176;

82 : -33;

83 : 451;

84 : 526;

85 : 468;

86 : 59;

87 : -57;

88 : -317;

89 : -241;

8a : -327;

8b : -366;

8c : -661;

8d : -859;

8e : -791;

8f : -878;

90 : -1149;

91 : -934;
```

```
92 : -787;

93 : -747;

94 : -841;

95 : -818;

96 : -746;

97 : -796;

98 : -597;

99 : -490;

9a : -534;

9b : -578;

9c : -265;

9d : -301;

9e : -119;

9f : -261;

a0 : -138;

a1 : -168;

a2 : -288;

a3 : -171;

a4 : -121;

a5 : -132;

a6 : -127;

a7 : -130;

a8 : -128;

a9 : -130;

aa : -129;

ab : -129;

ac : -129;

ad : -129;

ae : -129;
```

```
af : -129;
b0 : -129;
b1 : -129;
b2 : -130;
b3 : -128;
b4 : -131;
b5 : -126;
b6 : -133;
b7 : -122;
b8 : -143;
b9 : -70;
ba : 13;
bb : -8;
bc : 3;
bd : -4;
be : 1;
bf : -2;
c0 : 0;
c1 : -1;
c2 : -1;
c3 : -1;
c4 : -1;
c5 : -1;
c6 : -1;
c7 : -1;
c8 : -1;
c9 : -1;
ca : -1;
cb : -1;
```

```
cc : -1;
cd : -1;
ce : -1;
cf : -1;
d0 : -1;
d1 : -1;
d2 : -1;
d3 : -1;
d4 : -1;
d5 : -1;
d6 : -1;
d7 : -1;
d8 : -1;
d9 : -1;
da : -1;
db : -1;
dc : -1;
dd : -1;
de : -1;
df : -1;
e0 : -1;
e1 : -1;
e2 : -1;
e3 : -1;
e4 : -1;
e5 : -1;
e6 : -1;
e7 : -1;
e8 : -1;
```

```
e9 : -1;
ea : -1;
eb : 0;
ec : -2;
ed : 1;
ee : -4;
ef : 3;
f0 : -8;
f1 : 13;
f2 : -66;
f3 : -143;
f4 : -122;
f5 : -133;
f6 : -126;
f7 : -131;
f8 : -128;
f9 : -130;
fa : -128;
fb : -131;
fc : -126;
fd : -133;
fe : -122;
ff : -140;
100 : -108;
101 : -226;
102 : -198;
103 : -118;
104 : -278;
105 : -218;
```

```
106 : -116;
107 : -237;
108 : -263;
109 : -296;
10a : -309;
10b : -255;
10c : -287;
10d : -309;
10e : -275;
10f : -256;
110 : -336;
111 : -239;
112 : -456;
113 : -723;
114 : -744;
115 : -784;
116 : -759;
117 : -776;
118 : -765;
119 : -772;
11a : -767;
11b : -772;
11c : -759;
11d : -664;
11e : -815;
11f : -645;
120 : -415;
121 : -541;
122 : -493;
```

```
123 : -564;
124 : -551;
125 : -538;
126 : -484;
127 : -602;
128 : -448;
129 : -531;
12a : -506;
12b : -462;
12c : -445;
12d : -574;
12e : -548;
12f : -529;
130 : -369;
131 : -324;
132 : -247;
133 : -300;
134 : -304;
135 : -269;
136 : -245;
137 : -169;
138 : -258;
139 : -266;
13a : -315;
13b : -282;
13c : -265;
13d : -209;
13e : -113;
13f : -260;
```

```
140 : -240;
141 : -118;
142 : -211;
143 : -281;
144 : -123;
145 : -189;
146 : -235;
147 : -102;
148 : -256;
149 : -244;
14a : -131;
14b : -127;
14c : -239;
14d : -267;
14e : -294;
14f : -313;
150 : -242;
151 : -311;
152 : -199;
153 : -367;
154 : -445;
155 : -506;
156 : -513;
157 : -536;
158 : -569;
159 : -534;
15a : -506;
15b : -479;
15c : -557;
```

```
15d : -505;
15e : -568;
15f : -549;
160 : -527;
161 : -483;
162 : -433;
163 : -503;
164 : -118;
165 : -379;
166 : -86;
167 : 520;
168 : 599;
169 : 501;
16a : 498;
16b : -167;
16c : -281;
16d : -117;
16e : 2;
16f : -38;
170 : -102;
171 : -182;
172 : -318;
173 : -233;
174 : -288;
175 : -162;
176 : -364;
177 : -673;
178 : -793;
179 : -730;
```

```
17a : -1002;
17b : -1293;
17c : -1261;
17d : -1518;
17e : -1334;
17f : -1298;
180 : -1293;
181 : -1275;
182 : -1279;
183 : -1059;
184 : -986;
185 : -770;
186 : -769;
187 : -770;
188 : -765;
189 : -774;
18a : -759;
18b : -784;
18c : -740;
18d : -882;
18e : -1027;
18f : -1160;
190 : -1338;
191 : -1322;
192 : -1298;
193 : -1125;
194 : -1228;
195 : -979;
196 : -1053;
```

```
197 : -1033;

198 : -1030;

199 : -826;

19a : -756;

19b : -767;

19c : -852;

19d : -789;

19e : -760;

19f : -774;

1a0 : -766;

1a1 : -771;

1a2 : -768;

1a3 : -770;

1a4 : -768;

1a5 : -770;

1a6 : -767;

1a7 : -772;

1a8 : -764;

1a9 : -779;

1aa : -855;

1ab : -775;

1ac : -749;

1ad : -691;

1ae : -599;

1af : -350;

1b0 : -329;

1b1 : -249;

1b2 : -241;

1b3 : -107;
```

```
1b4 : -232;

1b5 : -266;

1b6 : -125;

1b7 : -185;

1b8 : -284;

1b9 : -161;

1ba : -122;

1bb : -132;

1bc : -127;

1bd : -130;

1be : -128;

1bf : -129;

1c0 : -129;

1c1 : -129;

1c2 : -129;

1c3 : -129;

1c4 : -129;

1c5 : -129;

1c6 : -129;

1c7 : -129;

1c8 : -129;

1c9 : -130;

1ca : -128;

1cb : -131;

1cc : -126;

1cd : -133;

1ce : -121;

1cf : -154;

1d0 : -256;
```

```
1d1 : -112;
1d2 : -213;
1d3 : -276;
1d4 : -138;
1d5 : -165;
1d6 : -289;
1d7 : -171;
1d8 : -135;
1d9 : -266;
1da : -294;
1db : -309;
1dc : -253;
1dd : -289;
1de : -308;
1df : -274;
1e0 : -258;
1e1 : -334;
1e2 : -242;
1e3 : -458;
1e4 : -828;
1e5 : -726;
1e6 : -817;
1e7 : -808;
1e8 : -936;
1e9 : -1070;
1ea : -834;
1eb : -838;
1ec : -741;
1ed : -795;
```

```
1ee : -891;
1ef : -842;
1f0 : -671;
1f1 : -745;
1f2 : -545;
1f3 : -705;
1f4 : -779;
1f5 : -510;
1f6 : -527;
1f7 : -491;
1f8 : -595;
1f9 : -455;
1fa : -518;
1fb : -593;
1fc : -481;
1fd : -442;
1fe : -573;
1ff : -556;
200 : -508;
201 : -442;
202 : -359;
203 : -267;
204 : -261;
205 : -158;
206 : 8;
207 : -6;
208 : 8;
209 : -54;
20a : -114;
```

```
20b : -169;

20c : -237;

20d : -268;

20e : -300;

20f : -303;

210 : -253;

211 : -294;

212 : -310;

213 : -263;

214 : -275;

215 : -318;

216 : -270;

217 : -266;

218 : -176;

219 : -128;

21a : -288;

21b : -196;

21c : -136;

21d : -171;

21e : -128;

21f : -115;

220 : -219;

221 : -253;

222 : -309;

223 : -194;

224 : -385;

225 : -508;

226 : -410;

227 : -637;
```

```
228 : -543;
229 : -580;
22a : -488;
22b : -594;
22c : -656;
22d : -483;
22e : -530;
22f : -556;
230 : -555;
231 : -526;
232 : -478;
233 : -443;
234 : -484;
235 : -142;
236 : -291;
237 : 271;
238 : 1131;
239 : 1586;
23a : 1643;
23b : 1145;
23c : 241;
23d : -284;
23e : -68;
23f : 398;
240 : 561;
241 : 190;
242 : -87;
243 : -188;
244 : -269;
```

```
245 : -254;
246 : -149;
247 : -427;
248 : -753;
249 : -855;
24a : -1173;
24b : -1519;
24c : -1591;
24d : -1608;
24e : -1599;
24f : -1572;
250 : -1615;
251 : -1528;
252 : -1538;
253 : -1543;
254 : -1525;
255 : -1561;
256 : -1479;
257 : -1497;
258 : -1349;
259 : -1013;
25a : -1018;
25b : -1295;
25c : -1459;
25d : -1479;
25e : -1460;
25f : -1541;
260 : -1531;
261 : -1548;
```

```
262 : -1516;
263 : -1618;
264 : -1568;
265 : -1537;
266 : -1503;
267 : -1455;
268 : -1572;
269 : -1433;
26a : -1537;
26b : -1471;
26c : -1178;
26d : -1336;
26e : -1305;
26f : -1234;
270 : -1318;
271 : -1365;
272 : -1293;
273 : -1294;
274 : -1342;
275 : -1300;
276 : -1284;
277 : -1340;
278 : -1308;
279 : -1296;
27a : -989;
27b : -1072;
27c : -774;
27d : -782;
27e : -741;
```

```
27f : -828;

280 : -436;

281 : -183;

282 : -280;

283 : -259;

284 : -190;

285 : -139;

286 : -140;

287 : -90;

288 : -148;

289 : -118;

28a : -136;

28b : -125;

28c : -131;

28d : -127;

28e : -130;

28f : -128;

290 : -130;

291 : -128;

292 : -130;

293 : -128;

294 : -131;

295 : -123;

296 : -165;

297 : -290;

298 : -170;

299 : -141;

29a : -248;

29b : -165;
```

```
29c : -269;

29d : -253;

29e : -203;

29f : -295;

2a0 : -283;

2a1 : -237;

2a2 : -40;

2a3 : 24;

2a4 : -47;

2a5 : -72;

2a6 : 16;

2a7 : -9;

2a8 : 3;

2a9 : -2;

2aa : -2;

2ab : 3;

2ac : -6;

2ad : -125;

2ae : -268;

2af : -277;

2b0 : -19;

2b1 : -175;

2b2 : -285;

2b3 : -139;

2b4 : -47;

2b5 : -340;

2b6 : -256;

2b7 : -260;

2b8 : -292;
```

```
2b9 : -317;
2ba : -241;
2bb : -616;
2bc : -805;
2bd : -748;
2be : -788;
2bf : -676;
2c0 : -755;
2c1 : -781;
2c2 : -754;
2c3 : -794;
2c4 : -708;
2c5 : -722;
2c6 : -716;
2c7 : -714;
2c8 : -792;
2c9 : -755;
2ca : -777;
2cb : -767;
2cc : -908;
2cd : -828;
2ce : -741;
2cf : -887;
2d0 : -860;
2d1 : -758;
2d2 : -771;
2d3 : -772;
2d4 : -756;
2d5 : -848;
```

```
2d6 : -801;
2d7 : -753;
2d8 : -781;
2d9 : -591;
2da : -753;
2db : -691;
2dc : -741;
2dd : -693;
2de : -756;
2df : -834;
2e0 : -819;
2e1 : -751;
2e2 : -779;
2e3 : -763;
2e4 : -773;
2e5 : -767;
2e6 : -870;
2e7 : -752;
2e8 : -790;
2e9 : -737;
2ea : -832;
2eb : -892;
2ec : -1199;
2ed : -1355;
2ee : -1188;
2ef : -1394;
2f0 : -1463;
2f1 : -1468;
2f2 : -1549;
```

```
2f3 : -1474;

2f4 : -1483;

2f5 : -1570;

2f6 : -1486;

2f7 : -1469;

2f8 : -1552;

2f9 : -1530;

2fa : -1540;

2fb : -1536;

2fc : -1541;

2fd : -1680;

2fe : -1584;

2ff : -1531;

300 : -1451;

301 : -1527;

302 : -1543;

303 : -1534;

304 : -1460;

305 : -1418;

306 : -1128;

307 : -887;

308 : -927;

309 : -1037;

30a : -865;

30b : -910;

30c : -818;

30d : -914;

30e : -812;

30f : -754;
```

```
310 : -792;
311 : -862;
312 : -737;
313 : -809;
314 : -577;
315 : -538;
316 : -487;
317 : -322;
318 : -248;
319 : -316;
31a : -103;
31b : 16;
31c : -174;
31d : 0;
31e : -91;
31f : -18;
320 : 11;
321 : 306;
322 : 171;
323 : 176;
324 : 196;
325 : -57;
326 : 89;
327 : 175;
328 : 293;
329 : 506;
32a : 684;
32b : 1217;
32c : 1708;
```

```
32d : 2255;

32e : 2100;

32f : 1822;

330 : 2441;

331 : 2175;

332 : 343;

333 : -419;

334 : 492;

335 : 678;

336 : 79;

337 : 114;

338 : 115;

339 : -238;

33a : -700;

33b : -862;

33c : -830;

33d : -1213;

33e : -834;

33f : -735;

340 : -936;

341 : -1521;

342 : -1218;

343 : -1335;

344 : -1201;

345 : -733;

346 : -796;

347 : -748;

348 : -793;

349 : -598;
```

```
34a : -405;

34b : -703;

34c : -439;

34d : -425;

34e : -171;

34f : -230;

350 : -181;

351 : -241;

352 : -484;

353 : -210;

354 : -246;

355 : -340;

356 : -537;

357 : -499;

358 : -539;

359 : -681;

35a : -750;

35b : -609;

35c : -549;

35d : -584;

35e : -727;

35f : -687;

360 : -727;

361 : -775;

362 : -774;

363 : -751;

364 : -690;

365 : -700;

366 : -361;
```

```
367 : -372;

368 : -212;

369 : -225;

36a : -182;

36b : -234;

36c : -166;

36d : -281;

36e : -187;

36f : -126;

370 : -259;

371 : -137;

372 : -124;

373 : -191;

374 : -141;

375 : -145;

376 : -388;

377 : -262;

378 : 46;

379 : -130;

37a : -256;

37b : -291;

37c : -242;

37d : -144;

37e : -122;

37f : -104;

380 : -144;

381 : -120;

382 : -135;

383 : -126;
```

```
384 : -131;
385 : -128;
386 : -129;
387 : -129;
388 : -129;
389 : -129;
38a : -129;
38b : -129;
38c : -129;
38d : -129;
38e : -129;
38f : -129;
390 : -129;
391 : -129;
392 : -129;
393 : -129;
394 : -129;
395 : -129;
396 : -129;
397 : -129;
398 : -129;
399 : -129;
39a : -129;
39b : -129;
39c : -129;
39d : -129;
39e : -129;
39f : -129;
3a0 : -128;
```

```
3a1 : -133;
3a2 : -250;
3a3 : -250;
3a4 : -100;
3a5 : -244;
3a6 : -176;
3a7 : -136;
3a8 : -269;
3a9 : -259;
3aa : -317;
3ab : -286;
3ac : -258;
3ad : -302;
3ae : -303;
3af : -257;
3b0 : -286;
3b1 : -316;
3b2 : -257;
3b3 : -284;
3b4 : -299;
3b5 : -302;
3b6 : -215;
3b7 : -335;
3b8 : -262;
3b9 : -489;
3ba : -1355;
3bb : -1130;
3bc : -963;
3bd : -733;
```

```
3be : -796;

3bf : -740;

3c0 : -850;

3c1 : -808;

3c2 : -755;

3c3 : -778;

3c4 : -763;

3c5 : -774;

3c6 : -769;

3c7 : -700;

3c8 : -264;

3c9 : -225;

3ca : -369;

3cb : -204;

3cc : -255;

3cd : -274;

3ce : -305;

3cf : -301;

3d0 : -216;

3d1 : -177;

3d2 : -126;

3d3 : -212;

3d4 : -262;

3d5 : -184;

3d6 : -108;

3d7 : -10;

3d8 : -156;

3d9 : -17;

3da : 12;
```

```
3db : -45;

3dc : -76;

3dd : 32;

3de : -115;

3df : -102;

3e0 : 26;

3e1 : -70;

3e2 : -135;

3e3 : 8;

3e4 : -27;

3e5 : -145;

3e6 : -28;

3e7 : 8;

3e8 : -130;

3e9 : -77;

3ea : 36;

3eb : -115;

3ec : -91;

3ed : -22;

3ee : -169;

3ef : -220;

3f0 : -249;

3f1 : -238;

3f2 : -294;

3f3 : -473;

3f4 : -507;

3f5 : -593;

3f6 : -530;

3f7 : -639;
```

```
3f8 : -790;
3f9 : -675;
3fa : -671;
3fb : -793;
3fc : -635;
3fd : -574;
3fe : -543;
3ff : -601;
400 : -619;
401 : -654;
402 : -658;
403 : -716;
404 : -801;
405 : -719;
406 : -700;
407 : -783;
408 : -763;
409 : -770;
40a : -779;
40b : -986;
40c : -940;
40d : -1038;
40e : -1022;
40f : -1349;
410 : -1305;
411 : -1327;
412 : -1322;
413 : -1197;
414 : -1168;
```

```
415 : -1060;

416 : -807;

417 : -852;

418 : -744;

419 : -811;

41a : -856;

41b : -678;

41c : -591;

41d : -543;

41e : -435;

41f : -172;

420 : -218;

421 : -265;

422 : -291;

423 : -307;

424 : -273;

425 : -181;

426 : -157;

427 : 613;

428 : 1748;

429 : 1974;

42a : 2043;

42b : 1711;

42c : 109;

42d : -1076;

42e : -219;

42f : 470;

430 : 89;

431 : -47;
```

```
432 : -259;
433 : -674;
434 : -791;
435 : -757;
436 : -837;
437 : -1303;
438 : -1560;
439 : -1534;
43a : -1536;
43b : -2033;
43c : -2361;
43d : -2264;
43e : -2358;
43f : -2034;
440 : -1676;
441 : -1786;
442 : -1707;
443 : -1636;
444 : -1494;
445 : -1606;
446 : -1593;
447 : -1516;
448 : -1556;
449 : -1509;
44a : -1467;
44b : -1504;
44c : -1462;
44d : -1291;
44e : -786;
```

```
44f : -937;

450 : -1087;

451 : -1032;

452 : -857;

453 : -973;

454 : -994;

455 : -933;

456 : -886;

457 : -798;

458 : -833;

459 : -814;

45a : -760;

45b : -867;

45c : -767;

45d : -654;

45e : -536;

45f : -426;

460 : -580;

461 : -547;

462 : -449;

463 : -403;

464 : -304;

465 : -307;

466 : -247;

467 : -190;

468 : -217;

469 : -277;

46a : -289;

46b : -312;
```

```
46c : -253;

46d : -257;

46e : -107;

46f : -225;

470 : -199;

471 : -116;

472 : -280;

473 : -216;

474 : -123;

475 : -131;

476 : -128;

477 : -130;

478 : -128;

479 : -130;

47a : -129;

47b : -129;

47c : -129;

47d : -129;

47e : -129;

47f : -129;

480 : -129;

481 : -130;

482 : -128;

483 : -131;

484 : -126;

485 : -133;

486 : -122;

487 : -143;

488 : -53;
```

```
489 : 12;
48a : -8;
48b : 3;
48c : -4;
48d : 1;
48e : -2;
48f : 0;
490 : -1;
491 : -1;
492 : -1;
493 : -1;
494 : -1;
495 : -1;
496 : -1;
497 : -1;
498 : -1;
499 : -1;
49a : -1;
49b : -1;
49c : -1;
49d : -1;
49e : -1;
49f : -1;
4a0 : -1;
4a1 : -1;
4a2 : -1;
4a3 : -1;
4a4 : -1;
4a5 : -1;
```

```
4a6 : -1;
4a7 : -1;
4a8 : -1;
4a9 : -1;
4aa : -1;
4ab : -1;
4ac : -1;
4ad : -1;
4ae : -1;
4af : -1;
4b0 : -1;
4b1 : -1;
4b2 : -1;
4b3 : -1;
4b4 : -1;
4b5 : -1;
4b6 : -1;
4b7 : -1;
4b8 : -1;
4b9 : -1;
4ba : -1;
4bb : -1;
4bc : -1;
4bd : -1;
4be : -1;
4bf : -1;
4c0 : -1;
4c1 : -1;
4c2 : -1;
```

```
4c3 : -1;
4c4 : -1;
4c5 : -1;
4c6 : -1;
4c7 : -1;
4c8 : -1;
4c9 : -1;
4ca : -1;
4cb : -1;
4cc : -1;
4cd : -1;
4ce : -1;
4cf : -1;
4d0 : -1;
4d1 : -1;
4d2 : -1;
4d3 : -1;
4d4 : -1;
4d5 : -1;
4d6 : -1;
4d7 : -1;
4d8 : -1;
4d9 : -1;
4da : -1;
4db : -1;
4dc : -1;
4dd : -1;
4de : -1;
4df : -1;
```

```
4e0 : -1;
4e1 : -1;
4e2 : -1;
4e3 : -1;
4e4 : -1;
4e5 : -1;
4e6 : -1;
4e7 : -1;
4e8 : -1;
4e9 : -1;
4ea : -1;
4eb : -1;
4ec : -1;
4ed : -1;
4ee : -1;
4ef : -1;
4f0 : -1;
4f1 : -1;
4f2 : -1;
4f3 : -1;
4f4 : -1;
4f5 : -1;
4f6 : -1;
4f7 : -1;
4f8 : -1;
4f9 : -1;
4fa : -1;
4fb : 0;
4fc : -2;
```

```
4fd : 0;
4fe : -3;
4ff : 3;
500 : -7;
501 : 10;
502 : -41;
503 : -142;
504 : -123;
505 : -133;
506 : -127;
507 : -131;
508 : -128;
509 : -130;
50a : -129;
50b : -130;
50c : -128;
50d : -130;
50e : -127;
50f : -132;
510 : -125;
511 : -135;
512 : -117;
513 : -166;
514 : -251;
515 : -105;
516 : -228;
517 : -285;
518 : -313;
519 : -272;
```

```
51a : -263;

51b : -178;

51c : -226;

51d : -277;

51e : -286;

51f : -323;

520 : -223;

521 : -372;

522 : -840;

523 : -758;

524 : -763;

525 : -924;

526 : -823;

527 : -739;

528 : -972;

529 : -948;

52a : -1004;

52b : -933;

52c : -957;

52d : -1299;

52e : -1239;

52f : -1235;

530 : -1100;

531 : -972;

532 : -961;

533 : -999;

534 : -918;

535 : -776;

536 : -766;
```

```
537 : -774;
538 : -758;
539 : -801;
53a : -852;
53b : -744;
53c : -792;
53d : -728;
53e : -693;
53f : -789;
540 : -755;
541 : -787;
542 : -708;
543 : -669;
544 : -448;
545 : -507;
546 : -337;
547 : -256;
548 : -210;
549 : -199;
54a : -215;
54b : -188;
54c : -324;
54d : -326;
54e : -76;
54f : 18;
550 : -13;
551 : 13;
552 : -30;
553 : -87;
```

```
554 : 25;

555 : -26;

556 : 138;

557 : 59;

558 : 38;

559 : 200;

55a : 44;

55b : -13;

55c : 42;

55d : 264;

55e : 293;

55f : 387;

560 : 246;

561 : 225;

562 : 400;

563 : 405;

564 : 492;

565 : 429;

566 : 548;

567 : 397;

568 : 525;

569 : 477;

56a : 250;

56b : 269;

56c : 275;

56d : 50;

56e : 68;

56f : -26;

570 : 96;
```

```
571 : 11;
572 : -17;
573 : 64;
574 : 52;
575 : -33;
576 : -66;
577 : -33;
578 : -153;
579 : -301;
57a : -121;
57b : -186;
57c : 243;
57d : 2022;
57e : 1330;
57f : -3473;
580 : -6485;
581 : -5216;
582 : -3950;
583 : -4204;
584 : -2432;
585 : 1130;
586 : 981;
587 : -2125;
588 : -4069;
589 : -4501;
58a : -2910;
58b : -651;
58c : 601;
58d : 293;
```

```
58e : -1036;
58f : -3065;
590 : -3534;
591 : -2559;
592 : -1566;
593 : -78;
594 : 1310;
595 : 1622;
596 : 680;
597 : 34;
598 : -133;
599 : -221;
59a : 147;
59b : 807;
59c : 1228;
59d : 1206;
59e : 1378;
59f : 2184;
5a0 : 3165;
5a1 : 3457;
5a2 : 3335;
5a3 : 2965;
5a4 : 2103;
5a5 : 911;
5a6 : 996;
5a7 : 1637;
5a8 : 1980;
5a9 : 1644;
5aa : 1128;
```

```
5ab : 598;
5ac : -140;
5ad : -258;
5ae : -317;
5af : -228;
5b0 : 76;
5b1 : -117;
5b2 : -219;
5b3 : -738;
5b4 : -1407;
5b5 : -1468;
5b6 : -890;
5b7 : -777;
5b8 : -557;
5b9 : -683;
5ba : -745;
5bb : -1208;
5bc : -1593;
5bd : -1566;
5be : -1616;
5bf : -1425;
5c0 : -1526;
5c1 : -1549;
5c2 : -1512;
5c3 : -1893;
5c4 : -2385;
5c5 : -2023;
5c6 : -1538;
5c7 : -1534;
```

```
5c8 : -1442;
5c9 : -1546;
5ca : -1533;
5cb : -1535;
5cc : -1436;
5cd : -1561;
5ce : -1476;
5cf : -1449;
5d0 : -1407;
5d1 : -1520;
5d2 : -1542;
5d3 : -1536;
5d4 : -1535;
5d5 : -1446;
5d6 : -1537;
5d7 : -1541;
5d8 : -1528;
5d9 : -1458;
5da : -1470;
5db : -1066;
5dc : -1159;
5dd : -972;
5de : -760;
5df : -875;
5e0 : -734;
5e1 : -792;
5e2 : -427;
5e3 : -296;
5e4 : -352;
```

```
5e5 : -600;
5e6 : -453;
5e7 : -423;
5e8 : -221;
5e9 : -277;
5ea : -276;
5eb : -238;
5ec : -138;
5ed : -158;
5ee : -201;
5ef : -265;
5f0 : -194;
5f1 : -123;
5f2 : -16;
5f3 : -89;
5f4 : 18;
5f5 : -9;
5f6 : -95;
5f7 : 12;
5f8 : -54;
5f9 : -356;
5fa : -243;
5fb : -264;
5fc : -300;
5fd : -301;
5fe : -267;
5ff : -151;
600 : -125;
601 : -129;
```

```
602 : -135;
603 : -25;
604 : 7;
605 : -5;
606 : 2;
607 : -3;
608 : 0;
609 : -2;
60a : -1;
60b : -1;
60c : -1;
60d : -1;
60e : -1;
60f : -1;
610 : -1;
611 : -1;
612 : -1;
613 : -1;
614 : -1;
615 : -1;
616 : -1;
617 : -1;
618 : -1;
619 : -1;
61a : -1;
61b : -1;
61c : -1;
61d : -1;
61e : -1;
```

```
61f : -1;
620 : -1;
621 : -1;
622 : -1;
623 : -1;
624 : -1;
625 : -1;
626 : -1;
627 : -1;
628 : -1;
629 : -1;
62a : -1;
62b : -1;
62c : -1;
62d : -1;
62e : -1;
62f : -1;
630 : -1;
631 : -1;
632 : -1;
633 : -1;
634 : -1;
635 : -1;
636 : -1;
637 : -1;
638 : -1;
639 : -1;
63a : -1;
63b : -1;
```

```
63c : -1;
63d : -1;
63e : -1;
63f : -1;
640 : -1;
641 : -1;
642 : -1;
643 : -1;
644 : -1;
645 : -1;
646 : -1;
647 : -1;
648 : -1;
649 : -1;
64a : -1;
64b : -1;
64c : -1;
64d : -1;
64e : -1;
64f : -1;
650 : -1;
651 : -1;
652 : -1;
653 : -1;
654 : -1;
655 : -1;
656 : -1;
657 : -1;
658 : -1;
```

```
659 : -1;
65a : -1;
65b : -1;
65c : -1;
65d : -1;
65e : -1;
65f : -1;
660 : -1;
661 : -1;
662 : -1;
663 : -1;
664 : -1;
665 : 0;
666 : -2;
667 : 1;
668 : -4;
669 : 3;
66a : -8;
66b : 12;
66c : -78;
66d : -144;
66e : -120;
66f : -137;
670 : -120;
671 : -144;
672 : -15;
673 : -82;
674 : -152;
675 : -168;
```

```
676 : -129;
677 : -273;
678 : -214;
679 : -179;
67a : -293;
67b : -142;
67c : -161;
67d : -265;
67e : -267;
67f : -314;
680 : -284;
681 : -253;
682 : -316;
683 : -282;
684 : -283;
685 : -198;
686 : -302;
687 : -230;
688 : -1053;
689 : -1283;
68a : -1300;
68b : -1328;
68c : -1159;
68d : -1215;
68e : -1098;
68f : -1082;
690 : -777;
691 : -810;
692 : -842;
```

```
693 : -750;
694 : -778;
695 : -764;
696 : -770;
697 : -771;
698 : -760;
699 : -675;
69a : -781;
69b : -759;
69c : -780;
69d : -755;
69e : -789;
69f : -740;
6a0 : -1071;
6a1 : -984;
6a2 : -715;
6a3 : -807;
6a4 : -729;
6a5 : -709;
6a6 : -580;
6a7 : -589;
6a8 : -287;
6a9 : -188;
6aa : -236;
6ab : -211;
6ac : -137;
6ad : -64;
6ae : -49;
6af : -48;
```

```
6b0 : -142;
6b1 : 5;
6b2 : 3;
6b3 : -57;
6b4 : -67;
6b5 : 53;
6b6 : -296;
6b7 : -273;
6b8 : -145;
6b9 : -120;
6ba : -40;
6bb : 5;
6bc : -5;
6bd : 5;
6be : -189;
6bf : -294;
6c0 : -93;
6c1 : 15;
6c2 : -6;
6c3 : -1;
6c4 : 4;
6c5 : -12;
6c6 : 20;
6c7 : -190;
6c8 : -206;
6c9 : -318;
6ca : -163;
6cb : 23;
6cc : -351;
```

```
6cd : -29;
6ce : 4;
6cf : -3;
6d0 : -7;
6d1 : -147;
6d2 : -40;
6d3 : 7;
6d4 : -1;
6d5 : -7;
6d6 : -165;
6d7 : -197;
6d8 : -287;
6d9 : -356;
6da : -383;
6db : -569;
6dc : -706;
6dd : -758;
6de : -770;
6df : -777;
6e0 : -748;
6e1 : -841;
6e2 : -807;
6e3 : -795;
6e4 : -851;
6e5 : -751;
6e6 : -709;
6e7 : -666;
6e8 : -738;
6e9 : -548;
```

```
6ea : -461;
6eb : -418;
6ec : -340;
6ed : -314;
6ee : 409;
6ef : 1932;
6f0 : 2539;
6f1 : 2397;
6f2 : 2418;
6f3 : 581;
6f4 : -1520;
6f5 : -937;
6f6 : 582;
6f7 : 163;
6f8 : -185;
6f9 : -61;
6fa : -546;
6fb : -956;
6fc : -956;
6fd : -1107;
6fe : -1473;
6ff : -1511;
700 : -1403;
701 : -1417;
702 : -1594;
703 : -2218;
704 : -2115;
705 : -1619;
706 : -1484;
```

```
707 : -1426;
708 : -1245;
709 : -1310;
70a : -855;
70b : -376;
70c : -269;
70d : -250;
70e : -209;
70f : -216;
710 : -251;
711 : -278;
712 : -308;
713 : -289;
714 : -225;
715 : -111;
716 : -239;
717 : -257;
718 : -135;
719 : -122;
71a : -232;
71b : -265;
71c : -119;
71d : -194;
71e : -283;
71f : -150;
720 : -151;
721 : -286;
722 : -188;
723 : -125;
```

```
724 : -259;
725 : -241;
726 : -102;
727 : -247;
728 : -170;
729 : -140;
72a : -274;
72b : -255;
72c : -327;
72d : -264;
72e : -340;
72f : -446;
730 : -486;
731 : -538;
732 : -440;
733 : -383;
734 : -521;
735 : -489;
736 : -362;
737 : -503;
738 : -428;
739 : -371;
73a : -392;
73b : -381;
73c : -388;
73d : -383;
73e : -387;
73f : -378;
740 : -246;
```

```
741 : -291;
742 : -395;
743 : -380;
744 : -388;
745 : -383;
746 : -386;
747 : -385;
748 : -385;
749 : -386;
74a : -383;
74b : -388;
74c : -378;
74d : -416;
74e : -536;
74f : -451;
750 : -365;
751 : -425;
752 : -528;
753 : -560;
754 : -554;
755 : -515;
756 : -493;
757 : -263;
758 : -257;
759 : -301;
75a : -304;
75b : -261;
75c : -249;
75d : -122;
```

```
75e : -197;
75f : -288;
760 : -132;
761 : -176;
762 : -241;
763 : -111;
764 : -138;
765 : -125;
766 : -130;
767 : -131;
768 : -124;
769 : -142;
76a : -47;
76b : 11;
76c : -7;
76d : 3;
76e : -3;
76f : 1;
770 : -2;
771 : 0;
772 : -1;
773 : -1;
774 : -1;
775 : -1;
776 : -1;
777 : -1;
778 : -1;
779 : -1;
77a : -1;
```

```
77b : -1;
77c : -1;
77d : -1;
77e : -1;
77f : -1;
780 : -1;
781 : -1;
782 : -1;
783 : -1;
784 : -1;
785 : -1;
786 : -1;
787 : -1;
788 : -1;
789 : -1;
78a : -1;
78b : -1;
78c : -1;
78d : -1;
78e : -1;
78f : -1;
790 : -1;
791 : -1;
792 : -1;
793 : -1;
794 : -1;
795 : -1;
796 : -1;
797 : -1;
```

```
798 : -1;
799 : -1;
79a : -1;
79b : -1;
79c : -1;
79d : -1;
79e : -1;
79f : -1;
7a0 : -1;
7a1 : -1;
7a2 : -1;
7a3 : -1;
7a4 : -1;
7a5 : -1;
7a6 : -1;
7a7 : -1;
7a8 : -1;
7a9 : -1;
7aa : -1;
7ab : -1;
7ac : -1;
7ad : -1;
7ae : -1;
7af : -1;
7b0 : -1;
7b1 : -1;
7b2 : -1;
7b3 : -1;
7b4 : -1;
```

```
7b5 : -1;

7b6 : -1;

7b7 : -1;

7b8 : -1;

7b9 : -1;

7ba : -1;

7bb : -1;

7bc : -1;

7bd : -1;

7be : -1;

7bf : -1;

7c0 : -1;

7c1 : -1;

7c2 : -1;

7c3 : -1;

7c4 : -1;

7c5 : -1;

7c6 : -1;

7c7 : -1;

7c8 : -1;

7c9 : -1;

7ca : -1;

7cb : -1;

7cc : -1;

7cd : -1;

7ce : -1;

7cf : -1;

7d0 : -1;

7d1 : -1;
```

```
7d2 : -1;
7d3 : -1;
7d4 : -1;
7d5 : -1;
7d6 : -1;
7d7 : -1;
7d8 : -1;
7d9 : -1;
7da : -1;
7db : -1;
7dc : -1;
7dd : -1;
7de : -1;
7df : -1;
7e0 : -1;
7e1 : -1;
7e2 : -1;
7e3 : -1;
7e4 : -1;
7e5 : -1;
7e6 : -1;
7e7 : -1;
7e8 : -1;
7e9 : -1;
7ea : -1;
7eb : -1;
7ec : -1;
7ed : -1;
7ee : -1;
```

```
7ef : -1;
7f0 : -1;
7f1 : -1;
7f2 : -1;
7f3 : -1;
7f4 : -1;
7f5 : -1;
7f6 : -1;
7f7 : -1;
7f8 : -1;
7f9 : -1;
7fa : -1;
7fb : -1;
7fc : -1;
7fd : -1;
7fe : -1;
7ff : -1;
800 : -1;
801 : -1;
802 : -1;
803 : -1;
804 : -1;
805 : -1;
806 : -1;
807 : -1;
808 : -1;
809 : -1;
80a : -1;
80b : -1;
```

```
80c : -1;
80d : -1;
80e : -1;
80f : -1;
810 : -1;
811 : -1;
812 : -1;
813 : -1;
814 : -1;
815 : -1;
816 : -1;
817 : -1;
818 : -1;
819 : -1;
81a : -1;
81b : -1;
81c : -1;
81d : -1;
81e : -1;
81f : -1;
820 : -1;
821 : -1;
822 : -1;
823 : -1;
824 : -1;
825 : -1;
826 : -1;
827 : -1;
828 : -1;
```

```
829 : -1;
82a : -1;
82b : -1;
82c : -1;
82d : -1;
82e : -1;
82f : -1;
830 : -1;
831 : -1;
832 : -1;
833 : -1;
834 : -1;
835 : 0;
836 : -2;
837 : 1;
838 : -4;
839 : 3;
83a : -8;
83b : 13;
83c : -74;
83d : -142;
83e : -122;
83f : -133;
840 : -126;
841 : -131;
842 : -128;
843 : -130;
844 : -129;
845 : -129;
```

```
846 : -129;
847 : -129;
848 : -129;
849 : -129;
84a : -129;
84b : -129;
84c : -129;
84d : -129;
84e : -129;
84f : -129;
850 : -128;
851 : -237;
852 : -263;
853 : -103;
854 : -229;
855 : -195;
856 : -123;
857 : -266;
858 : -257;
859 : -313;
85a : -291;
85b : -257;
85c : -298;
85d : -307;
85e : -258;
85f : -283;
860 : -315;
861 : -268;
862 : -269;
```

```
863 : -315;
864 : -281;
865 : -259;
866 : -307;
867 : -297;
868 : -255;
869 : -296;
86a : -307;
86b : -263;
86c : -272;
86d : -355;
86e : -585;
86f : -358;
870 : -268;
871 : -270;
872 : -191;
873 : -201;
874 : -317;
875 : -523;
876 : -654;
877 : -312;
878 : -340;
879 : -563;
87a : -134;
87b : -319;
87c : -495;
87d : -478;
87e : -507;
87f : -331;
```

```
880 : -748;

881 : -841;

882 : -471;

883 : -545;

884 : -562;

885 : -707;

886 : -421;

887 : -811;

888 : -705;

889 : -506;

88a : -788;

88b : -795;

88c : -949;

88d : -439;

88e : -541;

88f : -534;

890 : -435;

891 : -762;

892 : -860;

893 : -722;

894 : -566;

895 : -608;

896 : -889;

897 : -766;

898 : -806;

899 : -801;

89a : -683;

89b : -517;

89c : -663;
```

```
89d : -762;
89e : -591;
89f : -413;
8a0 : -902;
8a1 : -836;
8a2 : -617;
8a3 : -653;
8a4 : -516;
8a5 : -755;
8a6 : -517;
8a7 : -693;
8a8 : -775;
8a9 : -636;
8aa : -314;
8ab : -390;
8ac : -573;
8ad : -832;
8ae : -693;
8af : -460;
8b0 : -594;
8b1 : -803;
8b2 : -498;
8b3 : -490;
8b4 : -558;
8b5 : -611;
8b6 : -384;
8b7 : -515;
8b8 : -772;
8b9 : -653;
```

```
8ba : -1134;

8bb : -584;

8bc : -316;

8bd : -633;

8be : -762;

8bf : -770;

8c0 : -474;

8c1 : -738;

8c2 : -811;

8c3 : -396;

8c4 : -222;

8c5 : -1045;

8c6 : -801;

8c7 : -191;

8c8 : -584;

8c9 : -1685;

8ca : -1011;

8cb : -240;

8cc : -223;

8cd : -493;

8ce : -228;

8cf : -155;

8d0 : -1248;

8d1 : -752;

8d2 : -66;

8d3 : -92;

8d4 : -952;

8d5 : -1189;

8d6 : -472;
```

```
8d7 : -1528;

8d8 : -1344;

8d9 : -920;

8da : -495;

8db : -585;

8dc : -671;

8dd : -1157;

8de : -1174;

8df : -1618;

8e0 : -945;

8e1 : -99;

8e2 : -1186;

8e3 : -1127;

8e4 : -416;

8e5 : -321;

8e6 : -1146;

8e7 : -1039;

8e8 : -530;

8e9 : -1080;

8ea : -1208;

8eb : -918;

8ec : -295;

8ed : -661;

8ee : -1541;

8ef : -1031;

8f0 : -1242;

8f1 : -1988;

8f2 : -1785;

8f3 : -1668;
```

```
8f4 : -1476;

8f5 : -348;

8f6 : -1333;

8f7 : -1444;

8f8 : -984;

8f9 : -511;

8fa : -391;

8fb : -1521;

8fc : -1454;

8fd : -1810;

8fe : -1247;

8ff : -1452;

900 : -700;

901 : -14;

902 : -1895;

903 : -867;

904 : -1336;

905 : -2217;

906 : -2098;

907 : -936;

908 : -249;

909 : -2892;

90a : -1474;

90b : -1073;

90c : -842;

90d : -464;

90e : -423;

90f : -472;

910 : -2198;
```

```
911 : -195;

912 : -1569;

913 : -1361;

914 : -291;

915 : -1910;

916 : -139;

917 : -2319;

918 : -956;

919 : -401;

91a : -1034;

91b : -184;

91c : -1526;

91d : 951;

91e : -110;

91f : -332;

920 : 1511;

921 : -1080;

922 : -1435;

923 : -2740;

924 : -1696;

925 : -1081;

926 : -1020;

927 : -62;

928 : -100;

929 : -550;

92a : 185;

92b : 1014;

92c : -482;

92d : 450;
```

```
92e : -204;
92f : -227;
930 : 37;
931 : 389;
932 : -1388;
933 : -108;
934 : 176;
935 : -743;
936 : -112;
937 : -305;
938 : 1637;
939 : 151;
93a : -164;
93b : 135;
93c : -2277;
93d : -1740;
93e : 468;
93f : 224;
940 : -1759;
941 : -772;
942 : -2205;
943 : -385;
944 : -1239;
945 : -2576;
946 : 2151;
947 : 1703;
948 : -1054;
949 : -1500;
94a : -2474;
```

```
94b : -3851;

94c : -3479;

94d : -2167;

94e : -976;

94f : 2669;

950 : 1970;

951 : 1977;

952 : 632;

953 : -1584;

954 : -356;

955 : -2807;

956 : -647;

957 : -670;

958 : 943;

959 : 563;

95a : 3;

95b : 576;

95c : 776;

95d : 1946;

95e : -581;

95f : -597;

960 : 3758;

961 : 37;

962 : -1057;

963 : 3058;

964 : 44;

965 : 665;

966 : -1999;

967 : -775;
```

```
968 : 3662;
969 : -1324;
96a : -2005;
96b : -2610;
96c : 1409;
96d : 526;
96e : -3147;
96f : 464;
970 : -1763;
971 : -1984;
972 : -2158;
973 : 1804;
974 : 426;
975 : -3425;
976 : -3137;
977 : -1214;
978 : 1114;
979 : -878;
97a : -2592;
97b : 3780;
97c : 200;
97d : -3953;
97e : -2434;
97f : 411;
980 : -3356;
981 : -1430;
982 : 3209;
983 : 4229;
984 : 1414;
```

```
985 : -829;
986 : -16;
987 : -816;
988 : -3530;
989 : -1816;
98a : -1956;
98b : -2510;
98c : -3713;
98d : -4331;
98e : 1798;
98f : -122;
990 : -1017;
991 : 3620;
992 : 4916;
993 : 4015;
994 : 2672;
995 : 1784;
996 : -738;
997 : 2306;
998 : 5771;
999 : -745;
99a : -2586;
99b : -5421;
99c : -5203;
99d : -243;
99e : 3951;
99f : 5650;
9a0 : 1832;
9a1 : 4307;
```

```
9a2 : 4051;

9a3 : -1301;

9a4 : 4487;

9a5 : 4143;

9a6 : -3980;

9a7 : -2449;

9a8 : -2861;

9a9 : -192;

9aa : -800;

9ab : 1083;

9ac : 4953;

9ad : 22;

9ae : -3091;

9af : -2160;

9b0 : -785;

9b1 : 1876;

9b2 : -6608;

9b3 : -447;

9b4 : 6049;

9b5 : -4752;

9b6 : -354;

9b7 : 7093;

9b8 : 7991;

9b9 : -921;

9ba : -7977;

9bb : -7281;

9bc : -7613;

9bd : -1686;

9be : -1311;
```

```
9bf : -1058;

9c0 : 5424;

9c1 : -611;

9c2 : 980;

9c3 : 6037;

9c4 : 4965;

9c5 : 2735;

9c6 : -7968;

9c7 : -11183;

9c8 : -7455;

9c9 : 4981;

9ca : 8345;

9cb : 4200;

9cc : 4564;

9cd : -2181;

9ce : -2108;

9cf : -85;

9d0 : -655;

9d1 : -1989;

9d2 : -2795;

9d3 : -1057;

9d4 : -9917;

9d5 : -7184;

9d6 : -3427;

9d7 : 165;

9d8 : -1348;

9d9 : -402;

9da : 4827;

9db : -394;
```

```
9dc : -2499;

9dd : 1296;

9de : -1570;

9df : 5150;

9e0 : 4692;

9e1 : 3569;

9e2 : 1418;

9e3 : -6288;

9e4 : -8202;

9e5 : -6859;

9e6 : -4675;

9e7 : -2999;

9e8 : -3137;

9e9 : 2580;

9ea : 619;

9eb : -2631;

9ec : 3073;

9ed : 5301;

9ee : 5569;

9ef : -3500;

9f0 : -3435;

9f1 : -5016;

9f2 : -2362;

9f3 : -2665;

9f4 : 4172;

9f5 : 5936;

9f6 : 2698;

9f7 : -2274;

9f8 : -596;
```

```
9f9 : 4255;
9fa : 3678;
9fb : 2595;
9fc : 2474;
9fd : 2225;
9fe : -1418;
9ff : -7668;
a00 : -8707;
a01 : -7391;
a02 : 10945;
a03 : 6314;
a04 : 15812;
a05 : 5206;
a06 : -13890;
a07 : -14845;
a08 : -11814;
a09 : 8536;
a0a : 12164;
a0b : 5923;
a0c : -8111;
a0d : -15063;
a0e : -15741;
a0f : -10748;
a10 : -7500;
a11 : 12496;
a12 : 4932;
a13 : 5138;
a14 : -3692;
a15 : 3492;
```

```
a16 : 11785;
a17 : 662;
a18 : -11209;
a19 : -16135;
a1a : -7846;
a1b : -4637;
a1c : -2697;
a1d : 16451;
a1e : 13083;
a1f : 6277;
a20 : 10120;
a21 : -6889;
a22 : 6073;
a23 : 3983;
a24 : -4759;
a25 : -10181;
a26 : 2414;
a27 : -2777;
a28 : -2821;
a29 : 3447;
a2a : 15705;
a2b : 12744;
a2c : 13962;
a2d : 14105;
a2e : 12627;
a2f : 15757;
a30 : -280;
a31 : -2339;
a32 : -12560;
```

```
a33 : -11132;
a34 : -8633;
a35 : -15497;
a36 : -2308;
a37 : 11028;
a38 : 9290;
a39 : 15562;
a3a : 11761;
a3b : 15086;
a3c : -620;
a3d : 5375;
a3e : 3887;
a3f : -15021;
a40 : -12598;
a41 : -13340;
a42 : -4624;
a43 : 9997;
a44 : 11121;
a45 : 14261;
a46 : 2823;
a47 : -14285;
a48 : -6769;
a49 : 2729;
a4a : 9333;
a4b : 7375;
a4c : -7062;
a4d : -14240;
a4e : -5106;
a4f : -12018;
```

```
a50 : -1586;

a51 : 4159;

a52 : 528;

a53 : -10216;

a54 : -16858;

a55 : -11764;

a56 : -14629;

a57 : -6754;

a58 : -9732;

a59 : -15497;

a5a : -13241;

a5b : -6978;

a5c : -4195;

a5d : 9909;

a5e : -7194;

a5f : -14735;

a60 : 6582;

a61 : -7501;

a62 : -15555;

a63 : -15765;

a64 : -4989;

a65 : 3789;

a66 : -18643;

a67 : -3472;

a68 : 14402;

a69 : 13286;

a6a : 13675;

a6b : 13795;

a6c : 13127;
```

```
a6d : 14293;

a6e : 10807;

a6f : 16104;

a70 : 3934;

a71 : -1230;

a72 : 16366;

a73 : 12125;

a74 : 14535;

a75 : 13056;

a76 : 13819;

a77 : 13543;

a78 : 13356;

a79 : 14184;

a7a : 4917;

a7b : -11690;

a7c : -5238;

a7d : -12283;

a7e : -14918;

a7f : -14852;

a80 : -7404;

a81 : 11863;

a82 : 2594;

a83 : 6256;

a84 : 15544;

a85 : 1109;

a86 : -9632;

a87 : -11328;

a88 : -16451;

a89 : 291;
```

```
a8a : 14551;

a8b : 3665;

a8c : 2469;

a8d : -647;

a8e : -4233;

a8f : -13813;

a90 : -14881;

a91 : -14120;

a92 : -14061;

a93 : -15307;

a94 : -7392;

a95 : -11690;

a96 : -15900;

a97 : -10559;

a98 : 91;

a99 : -5243;

a9a : -15710;

a9b : -13282;

a9c : -2619;

a9d : -6150;

a9e : -14239;

a9f : -13031;

aa0 : -14174;

aa1 : 4336;

aa2 : 2512;

aa3 : -16471;

aa4 : 8340;

aa5 : 14640;

aa6 : 10360;
```

```
aa7 : 13311;

aa8 : 12826;

aa9 : 15212;

aaa : -1565;

aab : -8446;

aac : 5538;

aad : -12937;

aae : -14420;

aaf : -14329;

ab0 : -14628;

ab1 : -14669;

ab2 : -14619;

ab3 : -14739;

ab4 : -14378;

ab5 : -14992;

ab6 : -13870;

ab7 : -15618;

ab8 : -7776;

ab9 : 13371;

aba : 13464;

abb : 13531;

abc : 14282;

abd : 12931;

abe : 14509;

abf : -8409;

ac0 : 6724;

ac1 : 10107;

ac2 : -15604;

ac3 : -13944;
```

```
ac4 : -12818;

ac5 : -16281;

ac6 : -2056;

ac7 : 9884;

ac8 : 1100;

ac9 : 15179;

aca : 6138;

acb : -15077;

acc : -12746;

acd : -15687;

ace : -10774;

acf : -11685;

ad0 : -14783;

ad1 : -15506;

ad2 : -946;

ad3 : 1532;

ad4 : -2675;

ad5 : 3239;

ad6 : 6921;

ad7 : 11012;

ad8 : -10524;

ad9 : -15046;

ada : -14445;

adb : -14682;

adc : -14563;

add : -14739;

ade : -14580;

adf : -14160;

ae0 : -5529;
```

```
ae1 : -12740;
ae2 : -9859;
ae3 : 171;
ae4 : 4068;
ae5 : 4052;
ae6 : -4740;
ae7 : -7608;
ae8 : -691;
ae9 : -5442;
aea : -14203;
aeb : -13062;
aec : -14870;
aed : -14127;
aee : -12561;
aef : -10250;
af0 : 2199;
af1 : 6178;
af2 : 3832;
af3 : 1563;
af4 : -2204;
af5 : -5039;
af6 : -5348;
af7 : -14088;
af8 : -12180;
af9 : -460;
afa : -4925;
afb : -8749;
afc : -7165;
afd : -7054;
```

```
afe : -2618;
aff : -6224;
b00 : -14890;
b01 : -8636;
b02 : -2394;
b03 : -13379;
b04 : -14631;
b05 : -14803;
b06 : -13243;
b07 : -11879;
b08 : -15344;
b09 : -10769;
b0a : -7555;
b0b : -11952;
b0c : -9570;
b0d : -8280;
b0e : -10069;
b0f : -7866;
b10 : -8351;
b11 : -10121;
b12 : -5483;
b13 : -6456;
b14 : -3899;
b15 : 2565;
b16 : -1313;
b17 : -385;
b18 : 5434;
b19 : -6745;
b1a : -9435;
```

```
b1b : 2242;

b1c : 1348;

b1d : 2311;

b1e : 3168;

b1f : 8317;

b20 : 9258;

b21 : -335;

b22 : 5308;

b23 : 11185;

b24 : 2769;

b25 : 3838;

b26 : 3984;

b27 : -1764;

b28 : 853;

b29 : -1183;

b2a : -5380;

b2b : 864;

b2c : 3125;

b2d : 5587;

b2e : 1902;

b2f : -2312;

b30 : 1786;

b31 : 3702;

b32 : -5510;

b33 : -9978;

b34 : -5292;

b35 : -8936;

b36 : -12088;

b37 : -4885;
```

```
b38 : -8243;

b39 : -10850;

b3a : -8732;

b3b : -2580;

b3c : 484;

b3d : -5219;

b3e : -5000;

b3f : 2725;

b40 : 6055;

b41 : -7202;

b42 : -12770;

b43 : -6446;

b44 : -2523;

b45 : -2937;

b46 : -11877;

b47 : -7174;

b48 : -935;

b49 : -2081;

b4a : -3189;

b4b : -895;

b4c : 5611;

b4d : 4751;

b4e : 4435;

b4f : 6823;

b50 : 74;

b51 : -972;

b52 : 305;

b53 : 6354;

b54 : 4133;
```

```
b55 : -2456;
b56 : -1620;
b57 : -2176;
b58 : -6184;
b59 : -8525;
b5a : -5343;
b5b : 1303;
b5c : -1091;
b5d : -524;
b5e : -2079;
b5f : -555;
b60 : -5360;
b61 : -5568;
b62 : 4396;
b63 : 233;
b64 : -2245;
b65 : 5614;
b66 : -5760;
b67 : -11362;
b68 : -7505;
b69 : -7195;
b6a : -5881;
b6b : -2814;
b6c : -362;
b6d : 1849;
b6e : -4159;
b6f : -8059;
b70 : -3541;
b71 : -1007;
```

```
b72 : -11350;
b73 : -7531;
b74 : -259;
b75 : -5257;
b76 : -6622;
b77 : -3755;
b78 : -5636;
b79 : -5000;
b7a : -5780;
b7b : -7423;
b7c : -6524;
b7d : -4371;
b7e : -1370;
b7f : 1644;
b80 : -1215;
b81 : -2101;
b82 : -2802;
b83 : -2595;
b84 : -4533;
b85 : -3232;
b86 : -3418;
b87 : -1242;
b88 : -4045;
b89 : -7405;
b8a : -7310;
b8b : -6207;
b8c : -4100;
b8d : -1330;
b8e : -2424;
```

```
b8f : -2771;

b90 : -7304;

b91 : -7778;

b92 : -8531;

b93 : -10553;

b94 : -4278;

b95 : -1217;

b96 : -4256;

b97 : -120;

b98 : 2155;

b99 : -3187;

b9a : 465;

b9b : 3259;

b9c : 4587;

b9d : 8205;

b9e : 6169;

b9f : 2253;

ba0 : 5118;

ba1 : 1631;

ba2 : -368;

ba3 : 1200;

ba4 : -35;

ba5 : 1781;

ba6 : 3711;

ba7 : 3195;

ba8 : 4990;

ba9 : 4663;

baa : 6480;

bab : 8863;
```

```
bac : 6067;
bad : 6379;
bae : 7212;
baf : 4204;
bb0 : 3369;
bb1 : 4975;
bb2 : 3886;
bb3 : 2518;
bb4 : 2292;
bb5 : 167;
bb6 : 3487;
bb7 : 5224;
bb8 : 2558;
bb9 : 7147;
bba : 9497;
bbb : 5711;
bbc : 8506;
bbd : 8321;
bbe : 2217;
bbf : 8849;
bc0 : 9910;
bc1 : 847;
bc2 : 2392;
bc3 : 2030;
bc4 : -464;
bc5 : -164;
bc6 : 2476;
bc7 : 5857;
bc8 : 8214;
```

```
bc9 : 5913;
bca : 1206;
bcb : 3342;
bcc : 1116;
bcd : -2101;
bce : 1141;
bcf : 6148;
bd0 : 8681;
bd1 : 7537;
bd2 : 3063;
bd3 : 2816;
bd4 : 4022;
bd5 : 2186;
bd6 : -2282;
bd7 : 3744;
bd8 : 5038;
bd9 : 2703;
bda : 2066;
bdb : 2429;
bdc : 1666;
bdd : 280;
bde : 490;
bdf : 2836;
be0 : 2667;
be1 : 3627;
be2 : 4118;
be3 : 2886;
be4 : 853;
be5 : 261;
```

```
be6 : 4388;
be7 : 1197;
be8 : -651;
be9 : 336;
bea : 1964;
beb : 2202;
bec : -4633;
bed : -4891;
bee : 199;
bef : 2074;
bf0 : -2578;
bf1 : -3445;
bf2 : 530;
bf3 : -905;
bf4 : -4172;
bf5 : -936;
bf6 : 3400;
bf7 : 2551;
bf8 : -2428;
bf9 : -1216;
bfa : 2838;
bfb : -767;
bfc : -2691;
bfd : -319;
bfe : 2864;
bff : -368;
c00 : -3752;
c01 : -5283;
c02 : -3566;
```

```
c03 : -6026;

c04 : -4604;

c05 : 10;

c06 : -1361;

c07 : -5426;

c08 : 658;

c09 : -1279;

c0a : -5544;

c0b : -4878;

c0c : -3096;

c0d : -162;

c0e : -2366;

c0f : -6874;

c10 : -6137;

c11 : -3888;

c12 : -5995;

c13 : -7255;

c14 : -5442;

c15 : 288;

c16 : 736;

c17 : -1523;

c18 : -3682;

c19 : -124;

c1a : 245;

c1b : -2896;

c1c : -325;

c1d : 3372;

c1e : -963;

c1f : -3334;
```

```
c20 : 355;
c21 : -5466;
c22 : -7214;
c23 : -4767;
c24 : -6351;
c25 : -4001;
c26 : -2764;
c27 : -6022;
c28 : -6556;
c29 : -7470;
c2a : -8235;
c2b : -7964;
c2c : -8899;
c2d : -5751;
c2e : -3631;
c2f : -3970;
c30 : -19;
c31 : -3301;
c32 : -6812;
c33 : -7210;
c34 : -7199;
c35 : -7254;
c36 : -8374;
c37 : -6729;
c38 : -960;
c39 : -1074;
c3a : -2862;
c3b : -5349;
c3c : -5570;
```

```
c3d : -1429;

c3e : -2998;

c3f : -3820;

c40 : 928;

c41 : -3046;

c42 : -2746;

c43 : -2481;

c44 : -6894;

c45 : -8681;

c46 : -7118;

c47 : -4685;

c48 : -4080;

c49 : -2045;

c4a : -2685;

c4b : -7637;

c4c : -7226;

c4d : -6017;

c4e : -6123;

c4f : -2254;

c50 : -5403;

c51 : -4697;

c52 : -2345;

c53 : -4895;

c54 : -4373;

c55 : -5889;

c56 : -6634;

c57 : -4171;

c58 : -3628;

c59 : -7908;
```

```
c5a : -7809;

c5b : -2619;

c5c : -2579;

c5d : -5408;

c5e : -9328;

c5f : -6819;

c60 : -5088;

c61 : -5097;

c62 : -9404;

c63 : -6417;

c64 : -2036;

c65 : -3062;

c66 : 187;

c67 : -4291;

c68 : -5629;

c69 : -6514;

c6a : -3910;

c6b : -990;

c6c : -421;

c6d : -2306;

c6e : -4686;

c6f : -2811;

c70 : -2294;

c71 : -2661;

c72 : -3971;

c73 : -3019;

c74 : -4681;

c75 : -5148;

c76 : -5570;
```

```
c77 : -3976;
c78 : -4894;
c79 : -5780;
c7a : -4736;
c7b : -3095;
c7c : -4076;
c7d : -1760;
c7e : -2874;
c7f : -4137;
c80 : -3417;
c81 : -838;
c82 : 1371;
c83 : 1556;
c84 : 579;
c85 : -1292;
c86 : 2588;
c87 : 663;
c88 : -3871;
c89 : -3844;
c8a : -905;
c8b : -1751;
c8c : -2226;
c8d : -1076;
c8e : 68;
c8f : -1296;
c90 : -1658;
c91 : -987;
c92 : -543;
c93 : -203;
```

```
c94 : -4242;

c95 : -2779;

c96 : -300;

c97 : 314;

c98 : -3239;

c99 : -3331;

c9a : -2816;

c9b : -3012;

c9c : -42;

c9d : 1485;

c9e : -4122;

c9f : 362;

ca0 : 339;

ca1 : -1018;

ca2 : 84;

ca3 : -3953;

ca4 : -1799;

ca5 : 500;

ca6 : 1524;

ca7 : -779;

ca8 : -1618;

ca9 : -951;

caa : -2262;

cab : 787;

cac : 1079;

cad : 609;

cae : -1268;

caf : -3247;

cb0 : -829;
```

```
cb1 : 2114;

cb2 : -2813;

cb3 : -1397;

cb4 : 5241;

cb5 : 1242;

cb6 : 2240;

cb7 : -709;

cb8 : -173;

cb9 : 491;

cba : -484;

cbb : 3172;

cbc : -505;

cbd : -40;

cbe : -264;

cbf : -6305;

cc0 : -2088;

cc1 : 1163;

cc2 : -4530;

cc3 : -2711;

cc4 : -3894;

cc5 : 2855;

cc6 : 2322;

cc7 : 77;

cc8 : 2907;

cc9 : -9950;

cca : 3207;

ccb : -2300;

ccc : -843;

ccd : -1318;
```

```
cce : -3875;

ccf : -2489;

cd0 : -6627;

cd1 : 2375;

cd2 : -7066;

cd3 : -7431;

cd4 : -930;

cd5 : -5296;

cd6 : 7482;

cd7 : 608;

cd8 : -3412;

cd9 : 568;

cda : -171;

cdb : -1864;

cdc : -2631;

cdd : -7141;

cde : -7002;

cdf : -434;

ce0 : -4813;

ce1 : -1144;

ce2 : -1309;

ce3 : -3615;

ce4 : 51;

ce5 : -751;

ce6 : -3996;

ce7 : -1056;

ce8 : 382;

ce9 : -2098;

cea : 1969;
```

```
ceb : -1723;
cec : -5240;
ced : 242;
cee : -8304;
cef : -996;
cf0 : -5662;
cf1 : -5152;
cf2 : 1987;
cf3 : -2003;
cf4 : -2569;
cf5 : -1695;
cf6 : -4299;
cf7 : -2219;
cf8 : 365;
cf9 : -2395;
cfa : -2731;
cfb : -3418;
cfc : -1723;
cfd : 3307;
cfe : -2625;
cff : -3621;
d00 : -3972;
d01 : -4830;
d02 : -3309;
d03 : -2746;
d04 : -6480;
d05 : -4972;
d06 : -1718;
d07 : -264;
```

```
d08 : -2220;
d09 : -1806;
d0a : -2950;
d0b : -2916;
d0c : -1265;
d0d : 1771;
d0e : 1809;
d0f : -217;
d10 : -3129;
d11 : -2654;
d12 : 132;
d13 : -3747;
d14 : -6801;
d15 : -2577;
d16 : -4437;
d17 : 827;
d18 : 2535;
d19 : -3583;
d1a : 1058;
d1b : -2242;
d1c : -2097;
d1d : -453;
d1e : 85;
d1f : 1209;
d20 : -8;
d21 : 422;
d22 : -1016;
d23 : -2995;
d24 : 250;
```

```
d25 : -3965;
d26 : -2590;
d27 : -2966;
d28 : -3323;
d29 : -605;
d2a : -348;
d2b : -176;
d2c : -745;
d2d : -196;
d2e : 243;
d2f : -781;
d30 : -2535;
d31 : -4975;
d32 : -5257;
d33 : -2537;
d34 : -2735;
d35 : -2617;
d36 : -2312;
d37 : -1367;
d38 : -2304;
d39 : -3201;
d3a : -1259;
d3b : -5286;
d3c : -1521;
d3d : -1079;
d3e : -5518;
d3f : 1578;
d40 : -4474;
d41 : -1720;
```

```
d42 : -1961;
d43 : -4645;
d44 : -2662;
d45 : -4860;
d46 : -744;
d47 : -495;
d48 : -1202;
d49 : -2484;
d4a : -1160;
d4b : -1680;
d4c : -3611;
d4d : 777;
d4e : -3135;
d4f : -5088;
d50 : 719;
d51 : -2911;
d52 : -1272;
d53 : -1388;
d54 : -1873;
d55 : -3568;
d56 : 1399;
d57 : -1300;
d58 : -195;
d59 : -2865;
d5a : -1838;
d5b : 791;
d5c : -2575;
d5d : 102;
d5e : 1205;
```

```
d5f : -1223;
d60 : -2216;
d61 : -643;
d62 : -3182;
d63 : -712;
d64 : -952;
d65 : -2817;
d66 : -4454;
d67 : -65;
d68 : -889;
d69 : -2602;
d6a : -1213;
d6b : -2761;
d6c : -1532;
d6d : -532;
d6e : -1615;
d6f : -185;
d70 : -1945;
d71 : -2689;
d72 : -1027;
d73 : -3957;
d74 : -1363;
d75 : -1462;
d76 : -3551;
d77 : -194;
d78 : -2897;
d79 : 876;
d7a : -1977;
d7b : -2241;
```

```
d7c : -2103;

d7d : -1149;

d7e : -2150;

d7f : -2134;

d80 : 2247;

d81 : -2226;

d82 : 728;

d83 : 583;

d84 : -1492;

d85 : -2444;

d86 : -712;

d87 : -1869;

d88 : -2143;

d89 : -719;

d8a : 1203;

d8b : -1378;

d8c : -1490;

d8d : 942;

d8e : -3310;

d8f : -2921;

d90 : -1866;

d91 : -1863;

d92 : -1619;

d93 : -531;

d94 : -1065;

d95 : -870;

d96 : -1686;

d97 : -672;

d98 : -3406;
```

```
d99 : -2306;
d9a : -2766;
d9b : -998;
d9c : -1740;
d9d : -2237;
d9e : -991;
d9f : -3241;
da0 : -2238;
da1 : -1941;
da2 : -3135;
da3 : -2719;
da4 : -2724;
da5 : -2497;
da6 : -1212;
da7 : -1872;
da8 : -1526;
da9 : -2998;
daa : -1501;
dab : -2373;
dac : -1863;
dad : -2875;
dae : -1640;
daf : -2199;
db0 : -2326;
db1 : -1679;
db2 : -2078;
db3 : -290;
db4 : -1001;
db5 : -2097;
```

```
db6 : -1548;

db7 : -295;

db8 : -1188;

db9 : -1534;

dba : -2624;

dbb : -59;

dbc : -1459;

dbd : -1784;

dbe : -1563;

dbf : -1338;

dc0 : -1483;

dc1 : -1483;

dc2 : -538;

dc3 : -1272;

dc4 : -997;

dc5 : -322;

dc6 : -1427;

dc7 : -1127;

dc8 : -673;

dc9 : -1324;

dca : -875;

dcb : -1557;

dcc : -461;

dcd : -1103;

dce : -1157;

dcf : -1163;

dd0 : -545;

dd1 : -1379;

dd2 : -1348;
```

```
dd3 : 282;
dd4 : -595;
dd5 : -867;
dd6 : -316;
dd7 : -1465;
dd8 : -1178;
dd9 : -599;
dda : -692;
ddb : -651;
ddc : -167;
ddd : -189;
dde : -218;
ddf : -460;
de0 : -532;
de1 : -685;
de2 : -952;
de3 : -419;
de4 : -559;
de5 : -27;
de6 : -769;
de7 : -279;
de8 : -543;
de9 : -324;
dea : -599;
deb : -528;
dec : -603;
ded : -410;
dee : -142;
def : -490;
```

```
df0 : -702;
df1 : -284;
df2 : -737;
df3 : -529;
df4 : -225;
df5 : -130;
df6 : -548;
df7 : -595;
df8 : -541;
df9 : -458;
dfa : -486;
dfb : -335;
dfc : -576;
dfd : -510;
dfe : -324;
dff : -384;
e00 : -525;
e01 : -692;
e02 : -621;
e03 : -461;
e04 : -638;
e05 : -471;
e06 : -484;
e07 : -298;
e08 : -362;
e09 : -444;
e0a : -508;
e0b : -341;
e0c : -476;
```

```
e0d : -333;
e0e : -573;
e0f : -387;
e10 : -360;
e11 : -330;
e12 : -368;
e13 : -436;
e14 : -258;
e15 : -487;
e16 : -654;
e17 : -315;
e18 : -350;
e19 : -387;
e1a : -362;
e1b : -349;
e1c : -401;
e1d : -332;
e1e : -382;
e1f : -337;
e20 : -340;
e21 : -391;
e22 : -364;
e23 : -250;
e24 : -190;
e25 : -320;
e26 : -318;
e27 : -412;
e28 : -331;
e29 : -246;
```

```
e2a : -160;
e2b : -312;
e2c : -360;
e2d : -363;
e2e : -356;
e2f : -273;
e30 : -347;
e31 : -397;
e32 : -382;
e33 : -371;
e34 : -335;
e35 : -218;
e36 : -263;
e37 : -226;
e38 : -340;
e39 : -277;
e3a : -363;
e3b : -366;
e3c : -398;
e3d : -374;
e3e : -396;
e3f : -266;
e40 : -265;
e41 : -431;
e42 : -264;
e43 : -247;
e44 : -302;
e45 : -222;
e46 : -380;
```

```
e47 : -588;
e48 : -613;
e49 : -430;
e4a : -551;
e4b : -646;
e4c : -370;
e4d : -237;
e4e : -332;
e4f : -268;
e50 : -284;
e51 : -182;
e52 : -476;
e53 : -660;
e54 : -773;
e55 : -734;
e56 : -610;
e57 : -678;
e58 : -355;
e59 : -315;
e5a : -298;
e5b : -162;
e5c : -235;
e5d : -260;
e5e : -303;
e5f : -183;
e60 : -511;
e61 : -816;
e62 : -370;
e63 : -462;
```

```
e64 : -844;
e65 : -545;
e66 : -399;
e67 : -343;
e68 : -217;
e69 : -707;
e6a : -817;
e6b : -408;
e6c : -273;
e6d : -256;
e6e : -270;
e6f : -340;
e70 : -297;
e71 : -213;
e72 : -726;
e73 : -674;
e74 : -621;
e75 : -658;
e76 : -619;
e77 : -367;
e78 : -185;
e79 : -165;
e7a : -177;
e7b : -165;
e7c : -225;
e7d : -337;
e7e : -767;
e7f : -737;
e80 : -594;
```

```
e81 : -259;

e82 : -749;

e83 : -423;

e84 : -45;

e85 : -226;

e86 : -287;

e87 : -607;

e88 : -676;

e89 : -826;

e8a : -515;

e8b : -575;

e8c : -276;

e8d : -416;

e8e : -424;

e8f : -150;

e90 : -710;

e91 : -456;

e92 : -676;

e93 : -595;

e94 : -283;

e95 : -355;

e96 : -209;

e97 : -241;

e98 : -574;

e99 : -278;

e9a : -186;

e9b : -674;

e9c : -341;

e9d : -848;
```

```
e9e : -296;

e9f : -620;

ea0 : -172;

ea1 : -189;

ea2 : -352;

ea3 : -126;

ea4 : -304;

ea5 : -230;

ea6 : -695;

ea7 : -896;

ea8 : -674;

ea9 : -758;

eaa : -527;

eab : -216;

eac : -135;

ead : -199;

eae : -161;

eaf : -188;

eb0 : -838;

eb1 : -781;

eb2 : -635;

eb3 : -426;

eb4 : -360;

eb5 : 2;

eb6 : -57;

eb7 : -154;

eb8 : -171;

eb9 : -933;

eba : -763;
```

```
ebb : -806;
ebc : -577;
ebd : -150;
ebe : -456;
ebf : -61;
ec0 : 188;
ec1 : -143;
ec2 : -226;
ec3 : -228;
ec4 : -677;
ec5 : -973;
ec6 : -682;
ec7 : -419;
ec8 : -606;
ec9 : -300;
eca : -65;
ecb : -73;
ecc : -119;
ecd : -443;
ece : -971;
ecf : -666;
ed0 : -965;
ed1 : -736;
ed2 : -700;
ed3 : -287;
ed4 : -153;
ed5 : -104;
ed6 : -96;
ed7 : -352;
```

```
ed8 : -414;
ed9 : -686;
eda : -1099;
edb : -1104;
edc : -732;
edd : -726;
ede : -609;
edf : -256;
ee0 : -62;
ee1 : -3;
ee2 : -387;
ee3 : -67;
ee4 : -441;
ee5 : -845;
ee6 : -882;
ee7 : -840;
ee8 : -398;
ee9 : -708;
eea : -278;
eeb : -379;
eec : -426;
eed : -13;
eee : -143;
eef : -97;
ef0 : -659;
ef1 : -791;
ef2 : -848;
ef3 : -951;
ef4 : -811;
```

```
ef5 : -677;
ef6 : -308;
ef7 : -363;
ef8 : -745;
ef9 : -575;
efa : -460;
efb : -888;
efc : -879;
efd : -698;
efe : -69;
eff : -21;
f00 : -175;
f01 : 18;
f02 : -65;
f03 : -333;
f04 : -911;
f05 : -883;
f06 : -863;
f07 : -832;
f08 : -574;
f09 : -335;
f0a : -82;
f0b : 281;
f0c : 233;
f0d : -252;
f0e : -25;
f0f : -327;
f10 : -899;
f11 : -898;
```

```
f12 : -795;
f13 : -305;
f14 : -599;
f15 : -750;
f16 : -105;
f17 : -26;
f18 : -251;
f19 : -182;
f1a : -215;
f1b : -230;
f1c : -318;
f1d : -638;
f1e : -1157;
f1f : -163;
f20 : -634;
f21 : -477;
f22 : -391;
f23 : -387;
f24 : -316;
f25 : -657;
f26 : -145;
f27 : -192;
f28 : -727;
f29 : -681;
f2a : -737;
f2b : -162;
f2c : -266;
f2d : -799;
f2e : 45;
```

```
f2f : 92;
f30 : -806;
f31 : -480;
f32 : -624;
f33 : -516;
f34 : -604;
f35 : -775;
f36 : -541;
f37 : -221;
f38 : -276;
f39 : -135;
f3a : -124;
f3b : 90;
f3c : -438;
f3d : -1171;
f3e : -697;
f3f : -376;
f40 : -630;
f41 : -936;
f42 : -470;
f43 : 126;
f44 : -318;
f45 : -305;
f46 : -697;
f47 : -991;
f48 : -472;
f49 : -662;
f4a : -1032;
f4b : -651;
```

```
f4c : 177;

f4d : -323;

f4e : 54;

f4f : 52;

f50 : 17;

f51 : -333;

f52 : -658;

f53 : 79;

f54 : -205;

f55 : -508;

f56 : -241;

f57 : -365;

f58 : -117;

f59 : -428;

f5a : 49;

f5b : -255;

f5c : -709;

f5d : -582;

f5e : -656;

f5f : -699;

f60 : -629;

f61 : 6;

f62 : -185;

f63 : -293;

f64 : -406;

f65 : -297;

f66 : -287;

f67 : -537;

f68 : -898;
```

```
f69 : -442;
f6a : -620;
f6b : -647;
f6c : -751;
f6d : -711;
f6e : -490;
f6f : -94;
f70 : -114;
f71 : -208;
f72 : -265;
f73 : -281;
f74 : -612;
f75 : -743;
f76 : -458;
f77 : -475;
f78 : -759;
f79 : -401;
f7a : -845;
f7b : -500;
f7c : -274;
f7d : -270;
f7e : -205;
f7f : 13;
f80 : -196;
f81 : -123;
f82 : -236;
f83 : -349;
f84 : -595;
f85 : -468;
```

```
f86 : -925;
f87 : -920;
f88 : -434;
f89 : -369;
f8a : -392;
f8b : -386;
f8c : -340;
f8d : -340;
f8e : -542;
f8f : -482;
f90 : -288;
f91 : -129;
f92 : -294;
f93 : -357;
f94 : -340;
f95 : -818;
f96 : -780;
f97 : -188;
f98 : -542;
f99 : -886;
f9a : -592;
f9b : -306;
f9c : 36;
f9d : -28;
f9e : -194;
f9f : -217;
fa0 : -6;
fa1 : -633;
fa2 : -745;
```

```
fa3 : -578;
fa4 : -461;
fa5 : -716;
fa6 : -121;
fa7 : -287;
fa8 : 3;
fa9 : 622;
faa : -455;
fab : -349;
fac : -234;
fad : -527;
fae : -670;
faf : -259;
fb0 : -360;
fb1 : -740;
fb2 : -328;
fb3 : -551;
fb4 : -887;
fb5 : -779;
fb6 : -573;
fb7 : -675;
fb8 : -270;
fb9 : -143;
fba : -139;
fbb : -316;
fbc : -365;
fbd : -185;
fbe : -282;
fbf : -766;
```

```
fc0 : -831;

fc1 : -859;

fc2 : -740;

fc3 : -332;

fc4 : -310;

fc5 : -182;

fc6 : -300;

fc7 : 117;

fc8 : 122;

fc9 : -123;

fca : -624;

fcb : -691;

fcc : -867;

fcd : -779;

fce : -899;

fcf : -869;

fd0 : -766;

fd1 : -315;

fd2 : -663;

fd3 : -141;

fd4 : -40;

fd5 : -226;

fd6 : -212;

fd7 : -169;

fd8 : -604;

fd9 : -763;

fda : -678;

fdb : -769;

fdc : -372;
```

```
fdd : -150;

fde : -187;

fdf : -146;

fe0 : -136;

fe1 : 2;

fe2 : 58;

fe3 : -136;

fe4 : -338;

fe5 : -442;

fe6 : -937;

fe7 : -672;

fe8 : -682;

fe9 : -805;

fea : -697;

feb : -143;

fec : -585;

fed : -244;

fee : -63;

fef : -192;

ff0 : 125;

ff1 : -222;

ff2 : -267;

ff3 : -520;

ff4 : -701;

ff5 : -275;

ff6 : -744;

ff7 : -267;

ff8 : -401;

ff9 : -670;
```

```
ffa : -701;
ffb : -548;
ffc : -594;
ffd : -666;
ffe : -418;
fff : -644;
1000 : -625;
1001 : -652;
1002 : -68;
1003 : -178;
1004 : -210;
1005 : -211;
1006 : -285;
1007 : -282;
1008 : -585;
1009 : -431;
100a : -805;
100b : -606;
100c : -628;
100d : -613;
100e : -667;
100f : -346;
1010 : -204;
1011 : -236;
1012 : -573;
1013 : -333;
1014 : -289;
1015 : -147;
1016 : -284;
```

```
1017 : -330;

1018 : -306;

1019 : -643;

101a : -803;

101b : -695;

101c : -633;

101d : -640;

101e : -400;

101f : -370;

1020 : -205;

1021 : 67;

1022 : 57;

1023 : -29;

1024 : -145;

1025 : -245;

1026 : -181;

1027 : -593;

1028 : -851;

1029 : -811;

102a : -624;

102b : -701;

102c : -423;

102d : -244;

102e : -277;

102f : -410;

1030 : -113;

1031 : -349;

1032 : -560;

1033 : -438;
```

```
1034 : -801;
1035 : -455;
1036 : -179;
1037 : -607;
1038 : -444;
1039 : -338;
103a : -130;
103b : -121;
103c : -544;
103d : -556;
103e : -49;
103f : -356;
1040 : -389;
1041 : -414;
1042 : -219;
1043 : -552;
1044 : -827;
1045 : -426;
1046 : -535;
1047 : -551;
1048 : -362;
1049 : -350;
104a : -231;
104b : -104;
104c : -221;
104d : -193;
104e : -62;
104f : -375;
1050 : -628;
```

```
1051 : -718;

1052 : -793;

1053 : -754;

1054 : -772;

1055 : -414;

1056 : -383;

1057 : -367;

1058 : -160;

1059 : 31;

105a : -295;

105b : -334;

105c : -573;

105d : -471;

105e : -716;

105f : -783;

1060 : -369;

1061 : -160;

1062 : -611;

1063 : -297;

1064 : -167;

1065 : -430;

1066 : -324;

1067 : -445;

1068 : -610;

1069 : -650;

106a : -437;

106b : -384;

106c : -349;

106d : -357;
```

```
106e : 32;
106f : -431;
1070 : -467;
1071 : -152;
1072 : -587;
1073 : -664;
1074 : -648;
1075 : -618;
1076 : -421;
1077 : -277;
1078 : -270;
1079 : -179;
107a : -229;
107b : -179;
107c : -107;
107d : -187;
107e : -437;
107f : -719;
1080 : -421;
1081 : -327;
1082 : -118;
1083 : -187;
1084 : -52;
1085 : -170;
1086 : -692;
1087 : -625;
1088 : -599;
1089 : -708;
108a : -812;
```

```
108b : -565;
108c : -213;
108d : -185;
108e : -345;
108f : -234;
1090 : -45;
1091 : -284;
1092 : -674;
1093 : -329;
1094 : -262;
1095 : -234;
1096 : -406;
1097 : -261;
1098 : -203;
1099 : -273;
109a : -708;
109b : -693;
109c : -448;
109d : -800;
109e : -756;
109f : -375;
10a0 : -110;
10a1 : -398;
10a2 : -167;
10a3 : -20;
10a4 : -234;
10a5 : -371;
10a6 : -340;
10a7 : -534;
```

```
10a8 : -697;
10a9 : -588;
10aa : -663;
10ab : -240;
10ac : -510;
10ad : -234;
10ae : -141;
10af : -251;
10b0 : -405;
10b1 : -637;
10b2 : -638;
10b3 : -466;
10b4 : -621;
10b5 : -671;
10b6 : -298;
10b7 : -333;
10b8 : -155;
10b9 : -47;
10ba : -600;
10bb : -397;
10bc : -481;
10bd : -820;
10be : -527;
10bf : -259;
10c0 : -260;
10c1 : -445;
10c2 : -225;
10c3 : -121;
10c4 : -248;
```

```
10c5 : -531;
10c6 : -473;
10c7 : -415;
10c8 : -612;
10c9 : -640;
10ca : -534;
10cb : -357;
10cc : -178;
10cd : -34;
10ce : -130;
10cf : -265;
10d0 : -505;
10d1 : -514;
10d2 : -509;
10d3 : -774;
10d4 : -575;
10d5 : -373;
10d6 : -61;
10d7 : -209;
10d8 : -270;
10d9 : -191;
10da : -300;
10db : -591;
10dc : -615;
10dd : -338;
10de : -232;
10df : -208;
10e0 : -200;
10e1 : -132;
```

```
10e2 : -191;

10e3 : -630;

10e4 : -794;

10e5 : -745;

10e6 : -806;

10e7 : -475;

10e8 : -384;

10e9 : -181;

10ea : -81;

10eb : -173;

10ec : -143;

10ed : -250;

10ee : -389;

10ef : -660;

10f0 : -646;

10f1 : -789;

10f2 : -753;

10f3 : -617;

10f4 : -657;

10f5 : -199;

10f6 : -169;

10f7 : -130;

10f8 : -192;

10f9 : -142;

10fa : -321;

10fb : -400;

10fc : -382;

10fd : -374;

10fe : -526;
```

```
10ff : -725;
1100 : -161;
1101 : -478;
1102 : -548;
1103 : -301;
1104 : -269;
1105 : -287;
1106 : -472;
1107 : -283;
1108 : -154;
1109 : -71;
110a : -216;
110b : -518;
110c : -329;
110d : -256;
110e : -636;
110f : -813;
1110 : -481;
1111 : -258;
1112 : -533;
1113 : -460;
1114 : -194;
1115 : -199;
1116 : -313;
1117 : -180;
1118 : -277;
1119 : -432;
111a : -234;
111b : -253;
```

```
111c : -361;

111d : -649;

111e : -264;

111f : -325;

1120 : -406;

1121 : -598;

1122 : -405;

1123 : -72;

1124 : -267;

1125 : -142;

1126 : -333;

1127 : -377;

1128 : -335;

1129 : -446;

112a : -382;

112b : -394;

112c : -693;

112d : -387;

112e : -454;

112f : -679;

1130 : -514;

1131 : -224;

1132 : -451;

1133 : -389;

1134 : 9;

1135 : -229;

1136 : -418;

1137 : -268;

1138 : -264;
```

```
1139 : -370;
113a : -465;
113b : -290;
113c : -547;
113d : -478;
113e : -105;
113f : -304;
1140 : -184;
1141 : -187;
1142 : -108;
1143 : -344;
1144 : -211;
1145 : -185;
1146 : -462;
1147 : -705;
1148 : -369;
1149 : -536;
114a : -631;
114b : -349;
114c : -454;
114d : -369;
114e : -393;
114f : -287;
1150 : -199;
1151 : -171;
1152 : -122;
1153 : -253;
1154 : -371;
1155 : -316;
```

```
1156 : -269;
1157 : -141;
1158 : -376;
1159 : -152;
115a : -331;
115b : -367;
115c : -284;
115d : -286;
115e : -253;
115f : -426;
1160 : -364;
1161 : -393;
1162 : -261;
1163 : -249;
1164 : -219;
1165 : -190;
1166 : -391;
1167 : -696;
1168 : -409;
1169 : -412;
116a : -438;
116b : -407;
116c : -270;
116d : -129;
116e : -246;
116f : -395;
1170 : -98;
1171 : -150;
1172 : -382;
```

```
1173 : -433;
1174 : -570;
1175 : -694;
1176 : -430;
1177 : -332;
1178 : -428;
1179 : -248;
117a : -388;
117b : -366;
117c : -374;
117d : -376;
117e : -337;
117f : -406;
1180 : -327;
1181 : -139;
1182 : -96;
1183 : -164;
1184 : -248;
1185 : -400;
1186 : -379;
1187 : -385;
1188 : -590;
1189 : -591;
118a : -403;
118b : -215;
118c : -189;
118d : -185;
118e : -236;
118f : -159;
```

```
1190 : -343;

1191 : -639;

1192 : -636;

1193 : -648;

1194 : -623;

1195 : -404;

1196 : -372;

1197 : -273;

1198 : -190;

1199 : -161;

119a : -208;

119b : -310;

119c : -379;

119d : -408;

119e : -288;

119f : -103;

11a0 : -139;

11a1 : -128;

11a2 : -172;

11a3 : -206;

11a4 : -394;

11a5 : -375;

11a6 : -394;

11a7 : -416;

11a8 : -345;

11a9 : -381;

11aa : -239;

11ab : -298;

11ac : -405;
```

```
11ad : -154;
11ae : -153;
11af : -185;
11b0 : -155;
11b1 : -335;
11b2 : -306;
11b3 : -369;
11b4 : -399;
11b5 : -258;
11b6 : -360;
11b7 : -392;
11b8 : -380;
11b9 : -326;
11ba : -424;
11bb : -274;
11bc : -251;
11bd : -488;
11be : -157;
11bf : -356;
11c0 : -426;
11c1 : -346;
11c2 : -279;
11c3 : -106;
11c4 : -193;
11c5 : -112;
11c6 : -344;
11c7 : -457;
11c8 : -335;
11c9 : -372;
```

```
11ca : -385;
11cb : -390;
11cc : -170;
11cd : -241;
11ce : -157;
11cf : -116;
11d0 : -138;
11d1 : -132;
11d2 : -346;
11d3 : -374;
11d4 : -412;
11d5 : -331;
11d6 : -576;
11d7 : -589;
11d8 : -320;
11d9 : -166;
11da : -367;
11db : -280;
11dc : -192;
11dd : -210;
11de : -205;
11df : -289;
11e0 : -276;
11e1 : -332;
11e2 : -304;
11e3 : -196;
11e4 : -369;
11e5 : -216;
11e6 : -345;
```

```
11e7 : -386;
11e8 : -391;
11e9 : -372;
11ea : -203;
11eb : -390;
11ec : -390;
11ed : -352;
11ee : -243;
11ef : -256;
11f0 : -106;
11f1 : -222;
11f2 : -264;
11f3 : -157;
11f4 : -99;
11f5 : -307;
11f6 : -404;
11f7 : -377;
11f8 : -388;
11f9 : -368;
11fa : -146;
11fb : -305;
11fc : -387;
11fd : -334;
11fe : -406;
11ff : -371;
1200 : -398;
1201 : -371;
1202 : -405;
1203 : -333;
```

```
1204 : -87;
1205 : -408;
1206 : -306;
1207 : -122;
1208 : -185;
1209 : -272;
120a : -379;
120b : -380;
120c : -343;
120d : -371;
120e : -191;
120f : -430;
1210 : -121;
1211 : -100;
1212 : -175;
1213 : -257;
1214 : -362;
1215 : -391;
1216 : -381;
1217 : -170;
1218 : -174;
1219 : -74;
121a : -180;
121b : -401;
121c : -219;
121d : -269;
121e : -267;
121f : -434;
1220 : -316;
```

```
1221 : -389;
1222 : -402;
1223 : -455;
1224 : -337;
1225 : -324;
1226 : -433;
1227 : -252;
1228 : -179;
1229 : -278;
122a : -244;
122b : -306;
122c : -298;
122d : -352;
122e : -341;
122f : -411;
1230 : -353;
1231 : -465;
1232 : -194;
1233 : -318;
1234 : -277;
1235 : -96;
1236 : -158;
1237 : -226;
1238 : -341;
1239 : -379;
123a : -180;
123b : -329;
123c : -263;
123d : -236;
```

```
123e : -303;

123f : -270;

1240 : -287;

1241 : -448;

1242 : -361;

1243 : -392;

1244 : -397;

1245 : -211;

1246 : -202;

1247 : -281;

1248 : -239;

1249 : -375;

124a : -384;

124b : -410;

124c : -430;

124d : -219;

124e : -231;

124f : -365;

1250 : -374;

1251 : -391;

1252 : -385;

1253 : -375;

1254 : -334;

1255 : -213;

1256 : -382;

1257 : -388;

1258 : -236;

1259 : -103;

125a : -255;
```

```
125b : -386;
125c : -390;
125d : -262;
125e : -361;
125f : -390;
1260 : -389;
1261 : -360;
1262 : -349;
1263 : -405;
1264 : -363;
1265 : -351;
1266 : -375;
1267 : -257;
1268 : -386;
1269 : -384;
126a : -325;
126b : -363;
126c : -351;
126d : -236;
126e : -291;
126f : -179;
1270 : -217;
1271 : -256;
1272 : -312;
1273 : -69;
1274 : -263;
1275 : -159;
1276 : -358;
1277 : -343;
```

```
1278 : -241;
1279 : -242;
127a : -104;
127b : -236;
127c : -292;
127d : -413;
127e : -344;
127f : -283;
1280 : -328;
1281 : -174;
1282 : -400;
1283 : -237;
1284 : -210;
1285 : -400;
1286 : -371;
1287 : -344;
1288 : -384;
1289 : -260;
128a : -369;
128b : -395;
128c : -372;
128d : -407;
128e : -297;
128f : -306;
1290 : -353;
1291 : -385;
1292 : -262;
1293 : -267;
1294 : -153;
```

```
1295 : -339;
1296 : -377;
1297 : -325;
1298 : -362;
1299 : -396;
129a : -368;
129b : -352;
129c : -359;
129d : -194;
129e : -452;
129f : -392;
12a0 : -391;
12a1 : -373;
12a2 : -404;
12a3 : -355;
12a4 : -446;
12a5 : -308;
12a6 : -211;
12a7 : -382;
12a8 : -325;
12a9 : -340;
12aa : -267;
12ab : -265;
12ac : -285;
12ad : -446;
12ae : -260;
12af : -203;
12b0 : -363;
12b1 : -67;
```

```
12b2 : -251;
12b3 : -226;
12b4 : -356;
12b5 : -388;
12b6 : -219;
12b7 : -287;
12b8 : -274;
12b9 : -208;
12ba : -358;
12bb : -372;
12bc : -231;
12bd : -317;
12be : -345;
12bf : -348;
12c0 : -308;
12c1 : -124;
12c2 : -248;
12c3 : -442;
12c4 : -368;
12c5 : -391;
12c6 : -238;
12c7 : -298;
12c8 : -381;
12c9 : -402;
12ca : -327;
12cb : -161;
12cc : -256;
12cd : -368;
12ce : -349;
```

```
12cf : -376;
12d0 : -325;
12d1 : -162;
12d2 : -383;
12d3 : -397;
12d4 : -235;
12d5 : -171;
12d6 : -124;
12d7 : -281;
12d8 : -427;
12d9 : -304;
12da : -131;
12db : -419;
12dc : -321;
12dd : -302;
12de : -323;
12df : -155;
12e0 : -117;
12e1 : -191;
12e2 : -239;
12e3 : -81;
12e4 : -320;
12e5 : -254;
12e6 : -196;
12e7 : -306;
12e8 : -410;
12e9 : -198;
12ea : -247;
12eb : -439;
```

```
12ec : -335;

12ed : -464;

12ee : -276;

12ef : -301;

12f0 : -465;

12f1 : -320;

12f2 : -472;

12f3 : -284;

12f4 : -243;

12f5 : -312;

12f6 : -156;

12f7 : -276;

12f8 : -399;

12f9 : -238;

12fa : -319;

12fb : -339;

12fc : -104;

12fd : -294;

12fe : -341;

12ff : -425;

1300 : -409;

1301 : -252;

1302 : -364;

1303 : -165;

1304 : -430;

1305 : -360;

1306 : -165;

1307 : -161;

1308 : -77;
```

```
1309 : -115;
130a : -238;
130b : -437;
130c : -294;
130d : -110;
130e : -332;
130f : -354;
1310 : -402;
1311 : -365;
1312 : -354;
1313 : -365;
1314 : -353;
1315 : -367;
1316 : -346;
1317 : -409;
1318 : -316;
1319 : -374;
131a : -273;
131b : -180;
131c : -406;
131d : -375;
131e : -278;
131f : -340;
1320 : -415;
1321 : -350;
1322 : -440;
1323 : -217;
1324 : -160;
1325 : -431;
```

```
1326 : -369;

1327 : -353;

1328 : -369;

1329 : -388;

132a : -393;

132b : -363;

132c : -456;

132d : -224;

132e : -380;

132f : -413;

1330 : -368;

1331 : -396;

1332 : -259;

1333 : -360;

1334 : -457;

1335 : -327;

1336 : -268;

1337 : -125;

1338 : -199;

1339 : -350;

133a : -150;

133b : -162;

133c : -64;

133d : -145;

133e : -120;

133f : -134;

1340 : -126;

1341 : -131;

1342 : -129;
```

```
1343 : -129;

1344 : -130;

1345 : -127;

1346 : -133;

1347 : -123;

1348 : -141;

1349 : -85;

134a : 13;

134b : -8;

134c : 3;

134d : -4;

134e : 1;

134f : -2;

1350 : 0;

1351 : -1;

1352 : -1;

1353 : -1;

1354 : -1;

1355 : -1;

1356 : -1;

1357 : -1;

1358 : -1;

1359 : -1;

135a : -1;

135b : -1;

135c : -1;

135d : -1;

135e : -1;

135f : -1;
```

```
1360 : -1;
1361 : -1;
1362 : -1;
1363 : -1;
1364 : -1;
1365 : -1;
1366 : -1;
1367 : -1;
1368 : -1;
1369 : -1;
136a : -1;
136b : -1;
136c : -1;
136d : -1;
136e : -1;
136f : -1;
1370 : -1;
1371 : -1;
END;
```

colorPals.sv

```
module colorPals (
  input logic [1:0] palID,
  input logic [5:0] address,
  output logic [23:0] rgb_out
);


        always_comb begin
                rgb_out = 24'd0;
                if (palID==0) begin
                        case(address)
                                0: rgb_out = 24'h000000;
                                1: rgb_out = 24'h1d2b53;
                                2: rgb_out = 24'h7e2553;
                                3: rgb_out = 24'h008751;
                                4: rgb_out = 24'hab5236;
                                5: rgb_out = 24'h5f574f;
                                6: rgb_out = 24'hc2c3c7;
                                7: rgb_out = 24'hfff1e8;
                                8: rgb_out = 24'hff004d;
                                9: rgb_out = 24'hffa300;
                                10: rgb_out = 24'hffec27;
                                11: rgb_out = 24'h00e436;
                                12: rgb_out = 24'h29adff;
                                13: rgb_out = 24'h83769c;
                                14: rgb_out = 24'hff77a8;
                                15: rgb_out = 24'hffccaa;
                                16: rgb_out = 24'h291814;
                                17: rgb_out = 24'h111d35;
                                18: rgb_out = 24'h422136;
                                19: rgb_out = 24'h125359;
                                20: rgb_out = 24'h742f29;
                                21: rgb_out = 24'h49333b;
                                22: rgb_out = 24'ha28879;
                                23: rgb_out = 24'hf3ef7d;
                                24: rgb_out = 24'hbe1250;
                                25: rgb_out = 24'hff6c24;
                                26: rgb_out = 24'ha8e72e;
                                27: rgb_out = 24'h00b543;
                                28: rgb_out = 24'h065ab5;
                                29: rgb_out = 24'h754665;
                                30: rgb_out = 24'hff6e59;
                                31: rgb_out = 24'hff9d81;
                                default: rgb_out = 24'h000000;
                        endcase
                end

                else if (palID==1) begin
                        case(address)
                                0: rgb_out = 24'h000000;
                                1: rgb_out = 24'h000000;
                                2: rgb_out = 24'hffffff;
                                3: rgb_out = 24'h595a5b;
                                4: rgb_out = 24'h0c0c0c;
                                5: rgb_out = 24'hcacaca;
                                6: rgb_out = 24'ha8b3bc;
                                7: rgb_out = 24'hb0b9c7;
                                8: rgb_out = 24'h2f3133;
                                9: rgb_out = 24'hcc0000;
                                10: rgb_out = 24'h858b92;
                                11: rgb_out = 24'ha1c7c6;
                                12: rgb_out = 24'h8fa4a8;
                                13: rgb_out = 24'hd3e9e5;
                                14: rgb_out = 24'h86969a;
                                15: rgb_out = 24'h89c5bb;
                                16: rgb_out = 24'hd2d3d6;
                                17: rgb_out = 24'hbbc7d2;
                                18: rgb_out = 24'h323539;
                                19: rgb_out = 24'h96a0a9;
                                20: rgb_out = 24'h342d2e;
                                21: rgb_out = 24'h45494d;
                                22: rgb_out = 24'h2b2f31;
                                23: rgb_out = 24'h2c2f32;
                                24: rgb_out = 24'h141617;
                                25: rgb_out = 24'h151420;
                                26: rgb_out = 24'h610f1a;
                                27: rgb_out = 24'hff7149;
                                28: rgb_out = 24'h0b2458;
                                29: rgb_out = 24'h0c5c67;
                                30: rgb_out = 24'h12916b;
                                31: rgb_out = 24'h27e931;
                                32: rgb_out = 24'hfff34f;
                                33: rgb_out = 24'hf6c23b;
                                34: rgb_out = 24'he97b21;
                                35: rgb_out = 24'hd44b49;
                                36: rgb_out = 24'ha21839;
                                37: rgb_out = 24'h5d093a;
                                38: rgb_out = 24'h3e0346;
                                39: rgb_out = 24'h7d1475;
                                40: rgb_out = 24'hba2e89;
                                41: rgb_out = 24'hf481b0;
                                42: rgb_out = 24'heeb8b4;
                                43: rgb_out = 24'h9756c7;
                                44: rgb_out = 24'h2c226e;
                                45: rgb_out = 24'h11073a;
                                46: rgb_out = 24'h2424af;
                                47: rgb_out = 24'h4b7cdb;
                                48: rgb_out = 24'h6acaf4;
                                49: rgb_out = 24'h86ffed;
                                50: rgb_out = 24'hfff7e9;
                                51: rgb_out = 24'hffd8a5;
                                52: rgb_out = 24'hdd9c60;
                                53: rgb_out = 24'h752314;
                                54: rgb_out = 24'h4b050a;
                                55: rgb_out = 24'h2c0008;
                                56: rgb_out = 24'h35281f;
                                57: rgb_out = 24'h3c3c3c;
                                58: rgb_out = 24'h7f7f7f;
                                59: rgb_out = 24'hb8a7b9;
                                default: rgb_out = 24'h000000;
                            endcase
                        end

                        else if (palID==2) begin
                            case(address)
                                0: rgb_out = 24'h000000;
                                1: rgb_out = 24'h000000;
                                2: rgb_out = 24'hffffff;
                                3: rgb_out = 24'h595a5b;
                                4: rgb_out = 24'h0c0c0c;
                                5: rgb_out = 24'hcacaca;
```

```
6: rgb_out = 24'ha8b3bc;
7: rgb_out = 24'hb0b9c7;
8: rgb_out = 24'h2f3133;
9: rgb_out = 24'hf0f5f6;
10: rgb_out = 24'h858b92;
11: rgb_out = 24'ha1c7c6;
12: rgb_out = 24'h8fa4a8;
13: rgb_out = 24'hd3e9e5;
14: rgb_out = 24'h86969a;
15: rgb_out = 24'h89c5bb;
16: rgb_out = 24'hd2d3d6;
17: rgb_out = 24'hbbc7d2;
18: rgb_out = 24'h323539;
19: rgb_out = 24'h96a0a9;
20: rgb_out = 24'h342d2e;
21: rgb_out = 24'h45494d;
22: rgb_out = 24'h2b2f31;
23: rgb_out = 24'h2c2f32;
24: rgb_out = 24'h141617;
25: rgb_out = 24'h151420;
26: rgb_out = 24'h610f1a;
27: rgb_out = 24'hff7149;
28: rgb_out = 24'h0b2458;
29: rgb_out = 24'h0c5c67;
30: rgb_out = 24'h12916b;
31: rgb_out = 24'h27e931;
32: rgb_out = 24'hfff34f;
33: rgb_out = 24'hf6c23b;
34: rgb_out = 24'he97b21;
```

```
                35: rgb_out = 24'hd44b49;
                36: rgb_out = 24'ha21839;
                37: rgb_out = 24'h5d093a;
                38: rgb_out = 24'h3e0346;
                39: rgb_out = 24'h7d1475;
                40: rgb_out = 24'hba2e89;
                41: rgb_out = 24'hf481b0;
                42: rgb_out = 24'heeb8b4;
                43: rgb_out = 24'h9756c7;
                44: rgb_out = 24'h2c226e;
                45: rgb_out = 24'h11073a;
                46: rgb_out = 24'h2424af;
                47: rgb_out = 24'h4b7cdb;
                48: rgb_out = 24'h6acaf4;
                49: rgb_out = 24'h86ffed;
                50: rgb_out = 24'hfff7e9;
                51: rgb_out = 24'hffd8a5;
                52: rgb_out = 24'hdd9c60;
                53: rgb_out = 24'h752314;
                54: rgb_out = 24'h4b050a;
                55: rgb_out = 24'h2c0008;
                56: rgb_out = 24'h35281f;
                57: rgb_out = 24'h3c3c3c;
                58: rgb_out = 24'h7f7f7f;
                59: rgb_out = 24'hb8a7b9;
                default: rgb_out = 24'h000000;
            endcase
        end
```

```
else if (palID==3) begin

        case(address)

                0: rgb_out = 24'h000000;

                1: rgb_out = 24'h000000;

                2: rgb_out = 24'hffffff;

                3: rgb_out = 24'h595a5b;

                4: rgb_out = 24'h0c0c0c;

                5: rgb_out = 24'hcacaca;

                6: rgb_out = 24'ha8b3bc;

                7: rgb_out = 24'hb0b9c7;

                8: rgb_out = 24'h2f3133;

                9: rgb_out = 24'hf0f5f6;

                10: rgb_out = 24'h858b92;

                11: rgb_out = 24'h685a34;

                12: rgb_out = 24'hfcfc7f;

                13: rgb_out = 24'h9d8a34;

                14: rgb_out = 24'hb2a859;

                15: rgb_out = 24'hfefe6a;

                16: rgb_out = 24'hcebe3d;

                17: rgb_out = 24'h9d853a;

                18: rgb_out = 24'hfefe50;

                19: rgb_out = 24'h838383;

                20: rgb_out = 24'h4d4d4d;

                21: rgb_out = 24'ha3a3a3;

                22: rgb_out = 24'h4b79f1;

                23: rgb_out = 24'h1c598c;

                24: rgb_out = 24'he84af4;

                25: rgb_out = 24'had256c;

                26: rgb_out = 24'hc4c4c4;
```

```
27: rgb_out = 24'h010000;
28: rgb_out = 24'h482200;
29: rgb_out = 24'h1f0e00;
30: rgb_out = 24'h341700;
31: rgb_out = 24'h0c0500;
32: rgb_out = 24'h020100;
33: rgb_out = 24'h1f0d00;
34: rgb_out = 24'h602a00;
35: rgb_out = 24'h361509;
36: rgb_out = 24'hd95427;
37: rgb_out = 24'h7a2f16;
38: rgb_out = 24'hde6621;
39: rgb_out = 24'he3791c;
40: rgb_out = 24'h0d0502;
41: rgb_out = 24'hf6ce88;
42: rgb_out = 24'hed9e11;
43: rgb_out = 24'h434a5f;
44: rgb_out = 24'h11a5d4;
45: rgb_out = 24'h2cc5f6;
46: rgb_out = 24'he1240f;
47: rgb_out = 24'hf05a5a;
48: rgb_out = 24'h040200;
49: rgb_out = 24'h110700;
50: rgb_out = 24'h959ab1;
51: rgb_out = 24'hfc683b;
52: rgb_out = 24'hde4b1e;
53: rgb_out = 24'hdce1e7;
54: rgb_out = 24'hcfc6b8;
default: rgb_out = 24'h000000;
```

```
            endcase
        end
    end

endmodule
```

rr_audio_explode.hex

```
:020000040000FA
:2000000082838587888788878987898789868A816265483C413E403E403E403F403F3F452F
:2000200047494F4D4F4E4F4E4F4E4F4E504D586869807F807F807F807F80807F80819FA09F
:20004000B9C5C1C4C1C3C1C3C1C3C1C4BFCDD2DAE5E3E3E3E3E3E3E3E2E4E1E5DBC6C3AAA0
:20006000ACAAACAAACAAACAAABAAABA8766B3D2D342F3330333132323134 2E4E60779289D4
:200080008F8A8E8B8E8B8F8A90889CB8C7EAE5E7E6E6E7E5E7E5E8E4E9E0BBAB85808480AB
:2000A0008480848084808380838183868C8F9391939292929293919 48F977B6348313834373500
:2000C0003636363636373539414 74E4D4E4D4E4D4E4D4E4D4F4C516D809B9A9A9B9A9B99CD
:2000E0009B999B999B9A8D8173707270727072717271717171777F888B898A898A898A8AC9
:20010000898A898B857B706B6D6C6D6D6C6D6C6D6B6E6A748FA9B9B5B7B6B7B6B6B6B6B669
:20012000B6B6B6BBBEC2C0C1C0C1C0C1C0C1C0C1C0C1C6CED3D3D3D3D2D3D2D2D2D2D2D2EB
:20014000D2C7B7ACADACACACACACACACACACABAE95664646464646464646464646464 64ABE
:2001600053595959595959595959595A595A57679CC1C3C2C2C2C2C2C2C2C2C2C2C1C2C3C383
:20018000C3C3C3C3C3C3C3C3C3C3C2C4BB7D4C4B4B4B4B4B4C4B4C4C4C4C4C4E697F8080F7
:2001A000808080808080808080808082ACCECCCDCCCDCCCDCCCCCCCCCCCCCBBBAEAEAEAEAEC4
:2001C000AEAEAEAEAEAEAEADAEABA9A9A9A9A9A9A9A9A9A9A8A9A8A9917F81808180818008
:2001E000818081808180816C5F61606160616161616161616062544546454646464647FC
:2002000045474549426398959796979697979697969796979 79B9A9B9A9B9A9B9A9B9A9BA8
:200220009A9B99A0BDC0BFC0BFC0BFC0BFC0BEC0BEC0BB91868A878A87898789878987892E
```

```
:20024000878ABCCFC8CDC9CCC9CCC9CBCACACBC9CD6738483E453F44404441434342453EBD

:200260007FA49AA09B9F9C9E9D9D9D9D9E9CA088777B797A797A7A7A7A7A797B7792A8F8

:20028000A2A6A3A5A4A5A4A4A5A3A6A0AC74404C454A4749484849474A464C4175A59AA053

:2002A0009C9F9D9E9E9D9E9C9F9AA4795059555856575756585559535C4D89C8BBC2BEC0FF

:2002C000BFBFBFBFBFBEBFBEBFBDBBBBBBBBBBBBBABBBABBB9BCB8BEA78A8E8C8E8C8D8CF7

:2002E0008E8C8E8B8F899368323C363A3839383939393A393A383F484747474747474747E2

:2003000047484748474A4C4C4C4C4C4C4C4C4D4C4D4B4F486C938C908D8F8E8F8F8E8F8D79

:20032000908C936B464C494B4A4B4A4B4B4B4B4B4B4C3D303432333233323333333334C0

:20034000325E797377757675767576767676767676797B7A7B7A7B7A7B7A7B7A7B7A7B775121

:2003600042464346444644464446444743539CAFABADABADABADACADACACACACACACABACA2

:20038000ACABACABACABACABACABAC9F898787878787878787878888788879BB4B4B5B4B4D9

:2003A000B4B4B4B4B4B4B4B4B28E6E6F6E6F6E6F6E6F6F6F6F6F6E728692929292929292C9

:2003C00092929292291928D807B7B7B7B7B7B7B7B7B7B7B7B7B7B786F6B6C6C6C6C6C6C6CB4

:2003E0006C6C6C6B7CA4B7B6B6B6B6B6B6B6B6B6B6B5B7A4816E6F6F6F6F6F6F6F6F6F6F06

:200400006F6F685C555656565656565656565656565F6B727171727172727272717271724E

:200420007E8D969696969596959695979497927656414443434443454345434542485 96C18

:20044000777576767676767675767577747C8EA0ABA8AAA9A9A9A9A9A9A9A9A8A9A8A9A8A9A836

:20046000A9A8A9A9A9A9A8AAA6AD8E6E49373F3A3E3B3E3C3D3C3D3D3D40424546464646EB

:2004800046464646464746474747474848484848484848484948494845434141414141414141A3

:2004A00041414142414141414041414141404140424043 3D516071837E817F808080808089

:2004C00080807F8C929DA5A1A4A1A4A2A3A2A3A2A3A19793898688868888688868885C5

:2004E0008C9B9FB0B0B0B0B0B0AFB0AFB1AEB2A3978F7D817E807F807F7F807F807E93990C

:20050000A6B5B0B3B1B3B1B2B1B2B1B2B1ABABA5A4A4A5A4A5A3A5A3A6A1A8937B775456A5

:20052000545555555654565457536 96F788B888A88898989898989898985858381828181 1B

:200540008281828182818281 7F7F7E7E7E7E7E7E7E7E7D7E7D7E726F6C5F626061606160BB

:2005600061606260667E7A8E9A96999798979897989798999B9A9D9D9D9D9D9D9D9D9D9D97

:200580009D9D8B8B847678777777777777777777777767474727171717171717171717 1744B

:2005A0007474767777777777777777777776798D8D93A6A4A5A4A5A4A5A4A5A3A69B828977E2

:2005C0006568666766676668666865 7A8E86A1AFACADACADACADACADACAC8B8385625C5E6A
```

```
:2005E0005D5D5D5D5D5D5D5E5C5656554E4E4E4F4E4F4E4F4E4F4E545E5A616A69696969DC
:200600006969696969696A605A5D534D4E4E4E4E4E4E4E4E4F4D547779799CA4A2A3A2A3A2A223
:20062000A3A2A4967F877C646464646464646464656465605E5F5B5858585858585858595882
:20064000595D5E5D626363636363636363636363636363636363636363636463646265828B0C
:20066000849AAFABADACACACACADACADA9A5A4A59E9D9D9D9D9D9D9D9D9D9D877B81765FF0
:200680005F605F605F605F605F61656665696C6B6B6B6B6B6B6B6C666264625C5A5C5ADB
:2006A0005C5A5C5A5C5A5E696B68707A787979797979787977784908D8E9DA5A1A4A2A4A2B9
:2006C000A4A2A4A0959395908485858585858584867D7576766D666966686768676866626
:2006E0006A797B797D8D8D8D8D8C8D8C8E8C8F827A7B7C71676A686A696A696A696A6C6CCC
:200700006C6D6F6F6F6F6F6F6F6E706D818B898897A6A1A5A2A4A2A4A2A4A0979798968CB2
:200720008C8C8C8C8C8C8C8C8B8A8A8A8A888988898889888A878F9A98999BA9A8A9A9B6
:20074000A9A9A9A8A9A7888685886F6065626562656265617178A797F92909290919091 4D
:2007600090928B76787778635D5F5D5F5E5F5E5F5E707474727B8A878987898789878A83A9
:2007800076777678696466656665666566656A6B6B6B6C717170717071707170737C7B7C93
:2007A0007B82888687878787878787869095939494A1A1A1A1A1A0A1A0A29A80838085736081
:2007C0006562646264636462696C6C6C6C7374747474747474747474717171717 06D6E6D6E75
:2007E0006D6E6D6E6C838C898B8AA3A7A6A7A7A6A7A6A89E878C888D7F6B6E6D6E6D6E6D10
:200800006E6D83898788879EA3A2A2A2A2A2A1A39A8D918E91877B7D7C7D7C7D7C7D7C7976
:2008200079797979757575757575757574797D7C7D7C8085848584858485848 68A898A89FD
:200840008A8E8F8F8F8F8F8F8F8F807C7E7C7F726A6C6B6C6B6C6B6C6762636264615A5ABF
:200860005B5A5B5A5B5A5C5F605F605F636464646564656565647E8883888293A8A6A7A6A633
:20088000A6A6A6A5A4A4A4A4A4A4A2A3A2A3A2A3A2A39E8E8E8F8E8F81797B7A7B7A7B7A7BBF
:2008A0007C7D7D7D7D7E807F807F807F807E89928F918F919EA2A0A1A0A29FA398868787C3
:2008C0008689786C6F6D6F6D6E6D6F77797879777C8382838383838383818181818181 7FB3
:2008E0007F7F7F7F7F7F7F7E7D7E7D7E7D7D7B7C7B7C7B7D7A828F8E8F8F8D96A3A0A2A04C
:20090000A29FA39B807F807E81795E5E5E5E5E5E5E5E6A6D6C6C6C6D797B7A7B7A7B7A7B16
:200920007A7A7A7A7A7A797A797A797A797B726B6C6C6C6D665D5F5D5F5D5F5D65717070D3
:20094000706F748381828182818280767575757575 6B686968696869696767676767676662
:2009600065656565656565656465646564656364636463646365 6E6F6F6F6F6E767B797A5B
```

```
:20098000797A7A7980848383838282868D8C8C8C8C8C8C8C8C8C8C8C8C8D8D8D8D8D8D59
:2009A0008D8D8D8D8D8D8D8D8D8D8D8D8D8D8E8F8E8F8E8E8F9090908F918F9282797BA8
:2009C0007A7B7B79666665666566646E7E7B7D7B7D7B8C94919392939293979797979797F9
:2009E000999D9C9C9C9C9C9B89868787868883717271727172717E83828382838391939300D
:200A0000939393948F8A8C8B8B8B8C868384838483847F777978797879736C6E6D6E6D6EC8
:200A20006C69696969696A68656565656565656464646464646463626362636363637D8280BA
:200A4000828082809AA19FA09F9FA08D858885888589776C6F6E6E6E6E767C7A7C7A7C7A23
:200A6000818987888788878C8F8E8F8E8F8E91969595959595959595959595959595956D
:200A8000959594979A9A9A9A9A9A9A9F9F9F9F9EA09B9394939493939389878888888888B7
:200AA00089898989898989898A8A8A898A878080808080808079757776777757A8A8A8A8ACD
:200AC000898B88969F9D9E9D9E9C9A9A9A999A999A979596969696959191919191918E62
:200AE0008C8C8C8B8D897C7C7D7C7D7C7D7C7E756C6E6C6E6C707E7E7E7E7D7F7C84908E8F8E15
:200B00008F8D86868686868687837D7F7E7F7E808A8B8A8B8A8B8A8D97979797979796929192
:200B20009291929192918D8C8C8C8C8D827D7F7E7F7E7E7F747071707171716F6E6E6E6E6ED0
:200B40006E6E6D6B6C6B6C6B6E6F6F6F6F6F6F6F7173737373737373747474747474747475 84
:200B600075767576726C6D6D6D6D6D6D6C6466646664697372737373737275808080814D
:200B8000807E7E7E7E7E7E7E7E7B7B7B7B7B7B7B7B7B7B7B7B7B7B7B7B7B7B7B7B7E7D7ED0
:200BA0007D7E7D7E7D80808180817E7373737373737373746A64666566666A696A696A6A6AE0
:200BC000696C6E6D6E6D6E6F706F706F706F70707271727174828282828182808B93913A
:200BE00092929088898889888988889837E7F7E7F7F807F807F807F807F80808180817E778E
:200C0000787878787877797 36E706F6F6F706F706F706F706F70707070706F6D6D6D6D6DB2
:200C20006E6D6E6B6B6A6B69727C7A7C7A7B7A7C79858C8B8B8C82787B797A797A797B709D
:200C4000676A6869696A6A6A6A6A6A6A6A6B6B6B6C6A797E7C7D7D7D7D7E7C8A908E8F8EFC
:200C60008A8889888988898889848283828389 8A898A898A898A898F909090908F8F8F8F38
:200C80008F8F8F8F8F8D8E8D8E89828483848384838484817A797A797E828181818181818180C2
:200CA00083898889888B8B8B8B8B8B8B8B8B8C8E8E8E8E8D8381828182818281837A74757581
:200CC00074727272727272727272706E706E727D7C7C7C7C7C7C7C7C7C858B888B867B7B7BF9
:200CE0007B7B7B7C7B7C726C6E6D6F7070707070717071707373747279848283828383831D
:200D000083828D9391938C8586868686868686867D797B797F8483838383838383838A8D2D
```

```
:200D20008C8C8E8F8F8F8F8F8F8F8F90919191837E807E7F7F7F7F7F7E6F6D6E6E7D8016

:200D40007F807F807F807F81909191918D8E8D8E8D8E8D8E8D8D8A8A8A8B8D8D8D8D8D8C57

:200D60008D8C8D8D908E91887D7E7E7E7E7D7E7D7F786B6E6B757E7C7D7C7C7C7C7D7B838A

:200D80008D8B8C8B8C8B8B8B8B8C8B8C8B8B8B8B8A817F807F7F7F7F7F7F807773747477D0

:200DA0007777777777777777777A7A7A7B7C7C7C7C7C7D7C7D7C7D7E7F7E838685858585A8

:200DC00086858685878C8C838182818281818181827D767876767676767676767676767646

:200DE00075757479807F7F7F7F7F7F7F7F878A888B8B8B8B8B8B8B8B8B8B8D8D8D8C60

:200E00008B8B8B8B8B8B8B8B8B8A89898989898989898989898989898889888889827D7E7E7E7EE0

:200E20007E7E7E7E7E7573747778777877787878787878797B7C7B787979797979797979798F

:200E40007876767A7E7D7E7D7E7D7E7D7E7D828584868686868686868686868686868889868366

:200E6000848383838383838384827F7F808180818080808080808080808182817E7E7E7E7E7E5B

:200E80007E7E7E7E7E7B7B7A797979797979797979797A7978777B7F7E7F7E7E7E7E7E7F7EDF

:200EA000818584858585858585858585858585868585858585858484848484848484848280A6

:200EC0008080808080808080808080807D7D7F808080808080808080808081838383838383838300

:200EE00083838383838384838282828282828282828282808080807F7F7F7F7F7F7F7F7F7FCD

:200F00007F7F7F7E7E7E7E7E7E7E7E7E7E7E7E7E7E7D7D7D7D7D7D7D7D7D7D7D7D7D7D7D7D7E1E

:200F20007E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7F7F7F7FED

:200F40007F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F80808080808080808080808080A4

:200F6000808180808080808080808080808080808080808080808080808080808080808070

:200F8000808080808080808080808080808080808080808080808080808080808080808051

:200FA0008080808080807F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F4C

:200FC0007F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F7F807F8080808080802A

:200FE000808080808080808080808080808080808080808080808080807F807F80807F807FF5

:00000001FF
```

audioROM_fuel.v

```
// megafunction wizard: %ROM: 1-PORT%

// GENERATION: STANDARD

// VERSION: WM1.0

// MODULE: altsyncram


// ============================================================

// File Name: audioROM_fuel.v

// Megafunction Name(s):

//                     altsyncram

//

// Simulation Library Files(s):

//                     altera_mf

// ============================================================

// ************************************************************

// THIS IS A WIZARD-GENERATED FILE. DO NOT EDIT THIS FILE!

//

// 21.1.0 Build 842 10/21/2021 SJ Lite Edition

// ************************************************************



//Copyright (C) 2021  Intel Corporation. All rights reserved.

//Your use of Intel Corporation's design tools, logic functions

//and other software and tools, and any partner logic

//functions, and any output files from any of the foregoing

//(including device programming or simulation files), and any

//associated documentation or information are expressly subject

//to the terms and conditions of the Intel Program License

//Subscription Agreement, the Intel Quartus Prime License Agreement,
//the Intel FPGA IP License Agreement, or other applicable license
//agreement, including, without limitation, that your use is for
//the sole purpose of programming logic devices manufactured by
//Intel and sold by Intel or its authorized distributors.  Please
//refer to the applicable agreement for further details, at
//https://fpgasoftware.intel.com/eula.



// synopsys translate_off
`timescale 1 ps / 1 ps
// synopsys translate_on
module audioROM_fuel (
	address,
	clock,
	q);

	input	[10:0]  address;
	input	  clock;
	output	[7:0]  q;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_off
`endif
	tri1	  clock;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_on
`endif

	wire [7:0] sub_wire0;
	wire [7:0] q = sub_wire0[7:0];
	altsyncram	altsyncram_component (
				.address_a (address),
				.clock0 (clock),
				.q_a (sub_wire0),
				.aclr0 (1'b0),
				.aclr1 (1'b0),
				.address_b (1'b1),
				.addressstall_a (1'b0),
				.addressstall_b (1'b0),
				.byteena_a (1'b1),
				.byteena_b (1'b1),
				.clock1 (1'b1),
				.clocken0 (1'b1),
				.clocken1 (1'b1),
				.clocken2 (1'b1),
				.clocken3 (1'b1),
				.data_a ({8{1'b1}}),
				.data_b (1'b1),
				.eccstatus (),
				.q_b (),
				.rden_a (1'b1),
				.rden_b (1'b1),
				.wren_a (1'b0),
				.wren_b (1'b0));
	defparam
		altsyncram_component.address_aclr_a = "NONE",
		altsyncram_component.clock_enable_input_a = "BYPASS",
		altsyncram_component.clock_enable_output_a = "BYPASS",
`ifdef NO_PLI
		altsyncram_component.init_file = "rr_audio_fuel.rif"
`else
		altsyncram_component.init_file = "rr_audio_fuel.hex"
`endif
,
		altsyncram_component.intended_device_family = "Cyclone V",
		altsyncram_component.lpm_hint = "ENABLE_RUNTIME_MOD=NO",
		altsyncram_component.lpm_type = "altsyncram",
		altsyncram_component.numwords_a = 2048,
		altsyncram_component.operation_mode = "ROM",
		altsyncram_component.outdata_aclr_a = "NONE",
		altsyncram_component.outdata_reg_a = "UNREGISTERED",
		altsyncram_component.widthad_a = 11,
		altsyncram_component.width_a = 8,
		altsyncram_component.width_byteena_a = 1;


endmodule

// ============================================================
// CNX file retrieval info
// ============================================================
// Retrieval info: PRIVATE: ADDRESSSTALL_A NUMERIC "0"
// Retrieval info: PRIVATE: AclrAddr NUMERIC "0"
// Retrieval info: PRIVATE: AclrByte NUMERIC "0"
// Retrieval info: PRIVATE: AclrOutput NUMERIC "0"
// Retrieval info: PRIVATE: BYTE_ENABLE NUMERIC "0"
// Retrieval info: PRIVATE: BYTE_SIZE NUMERIC "8"
// Retrieval info: PRIVATE: BlankMemory NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_INPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_OUTPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: Clken NUMERIC "0"
// Retrieval info: PRIVATE: IMPLEMENT_IN_LES NUMERIC "0"
// Retrieval info: PRIVATE: INIT_FILE_LAYOUT STRING "PORT_A"
// Retrieval info: PRIVATE: INIT_TO_SIM_X NUMERIC "0"
// Retrieval info: PRIVATE: INTENDED_DEVICE_FAMILY STRING "Cyclone V"
// Retrieval info: PRIVATE: JTAG_ENABLED NUMERIC "0"
// Retrieval info: PRIVATE: JTAG_ID STRING "NONE"
// Retrieval info: PRIVATE: MAXIMUM_DEPTH NUMERIC "0"
// Retrieval info: PRIVATE: MIFfilename STRING "rr_audio_fuel.hex"
// Retrieval info: PRIVATE: NUMWORDS_A NUMERIC "2048"
// Retrieval info: PRIVATE: RAM_BLOCK_TYPE NUMERIC "0"
// Retrieval info: PRIVATE: RegAddr NUMERIC "1"
// Retrieval info: PRIVATE: RegOutput NUMERIC "0"
// Retrieval info: PRIVATE: SYNTH_WRAPPER_GEN_POSTFIX STRING "0"
// Retrieval info: PRIVATE: SingleClock NUMERIC "1"
// Retrieval info: PRIVATE: UseDQRAM NUMERIC "0"
// Retrieval info: PRIVATE: WidthAddr NUMERIC "11"
// Retrieval info: PRIVATE: WidthData NUMERIC "8"
// Retrieval info: PRIVATE: rden NUMERIC "0"
// Retrieval info: LIBRARY: altera_mf altera_mf.altera_mf_components.all
// Retrieval info: CONSTANT: ADDRESS_ACLR_A STRING "NONE"
// Retrieval info: CONSTANT: CLOCK_ENABLE_INPUT_A STRING "BYPASS"
// Retrieval info: CONSTANT: CLOCK_ENABLE_OUTPUT_A STRING "BYPASS"
// Retrieval info: CONSTANT: INIT_FILE STRING "rr_audio_fuel.hex"
// Retrieval info: CONSTANT: INTENDED_DEVICE_FAMILY STRING "Cyclone V"
// Retrieval info: CONSTANT: LPM_HINT STRING "ENABLE_RUNTIME_MOD=NO"
// Retrieval info: CONSTANT: LPM_TYPE STRING "altsyncram"

// Retrieval info: CONSTANT: NUMWORDS_A NUMERIC "2048"

// Retrieval info: CONSTANT: OPERATION_MODE STRING "ROM"

// Retrieval info: CONSTANT: OUTDATA_ACLR_A STRING "NONE"

// Retrieval info: CONSTANT: OUTDATA_REG_A STRING "UNREGISTERED"

// Retrieval info: CONSTANT: WIDTHAD_A NUMERIC "11"

// Retrieval info: CONSTANT: WIDTH_A NUMERIC "8"

// Retrieval info: CONSTANT: WIDTH_BYTEENA_A NUMERIC "1"

// Retrieval info: USED_PORT: address 0 0 11 0 INPUT NODEFVAL "address[10..0]"

// Retrieval info: USED_PORT: clock 0 0 0 0 INPUT VCC "clock"

// Retrieval info: USED_PORT: q 0 0 8 0 OUTPUT NODEFVAL "q[7..0]"

// Retrieval info: CONNECT: @address_a 0 0 11 0 address 0 0 11 0

// Retrieval info: CONNECT: @clock0 0 0 0 0 clock 0 0 0 0

// Retrieval info: CONNECT: q 0 0 8 0 @q_a 0 0 8 0

// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fuel.v TRUE

// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fuel.inc FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fuel.cmp FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fuel.bsf FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fuel_inst.v FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_fuel_bb.v TRUE

// Retrieval info: LIB_FILE: altera_mf
```

spriteRAM.v

```
// megafunction wizard: %RAM: 1-PORT%

// GENERATION: STANDARD
// VERSION: WM1.0
// MODULE: altsyncram

// ============================================================
// File Name: spriteRAM.v
// Megafunction Name(s):
//                      altsyncram
//
// Simulation Library Files(s):
//                      altera_mf
// ============================================================
// ************************************************************
// THIS IS A WIZARD-GENERATED FILE. DO NOT EDIT THIS FILE!
//
// 21.1.0 Build 842 10/21/2021 SJ Lite Edition
// ************************************************************


//Copyright (C) 2021  Intel Corporation. All rights reserved.
//Your use of Intel Corporation's design tools, logic functions
//and other software and tools, and any partner logic
//functions, and any output files from any of the foregoing
//(including device programming or simulation files), and any
//associated documentation or information are expressly subject
//to the terms and conditions of the Intel Program License
//Subscription Agreement, the Intel Quartus Prime License Agreement,
//the Intel FPGA IP License Agreement, or other applicable license
//agreement, including, without limitation, that your use is for
//the sole purpose of programming logic devices manufactured by
```

```
//Intel and sold by Intel or its authorized distributors.  Please
//refer to the applicable agreement for further details, at
//https://fpgasoftware.intel.com/eula.



// synopsys translate_off
`timescale 1 ps / 1 ps
// synopsys translate_on
module spriteRAM (
	address,
	clock,
	data,
	wren,
	q);

	input	[4:0]  address;
	input	  clock;
	input	[25:0]  data;
	input	  wren;
	output	[25:0]  q;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_off
`endif
	tri1	  clock;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_on
`endif

	wire [25:0] sub_wire0;
	wire [25:0] q = sub_wire0[25:0];


	altsyncram	altsyncram_component (
				.address_a (address),
				.clock0 (clock),
				.data_a (data),
				.wren_a (wren),
				.q_a (sub_wire0),
				.aclr0 (1'b0),
				.aclr1 (1'b0),
				.address_b (1'b1),
				.addressstall_a (1'b0),
				.addressstall_b (1'b0),
				.byteena_a (1'b1),
				.byteena_b (1'b1),
				.clock1 (1'b1),
				.clocken0 (1'b1),
				.clocken1 (1'b1),
				.clocken2 (1'b1),
				.clocken3 (1'b1),
				.data_b (1'b1),
				.eccstatus (),
				.q_b (),
				.rden_a (1'b1),
				.rden_b (1'b1),
				.wren_b (1'b0));
	defparam
		altsyncram_component.clock_enable_input_a = "BYPASS",
		altsyncram_component.clock_enable_output_a = "BYPASS",
                altsyncram_component.intended_device_family = "Cyclone V",

                altsyncram_component.lpm_hint = "ENABLE_RUNTIME_MOD=NO",

                altsyncram_component.lpm_type = "altsyncram",

                altsyncram_component.numwords_a = 32,

                altsyncram_component.operation_mode = "SINGLE_PORT",

                altsyncram_component.outdata_aclr_a = "NONE",

                altsyncram_component.outdata_reg_a = "UNREGISTERED",

                altsyncram_component.power_up_uninitialized = "FALSE",

                altsyncram_component.read_during_write_mode_port_a =
"NEW_DATA_NO_NBE_READ",

                altsyncram_component.widthad_a = 5,

                altsyncram_component.width_a = 26,

                altsyncram_component.width_byteena_a = 1;



endmodule


// ============================================================

// CNX file retrieval info

// ============================================================

// Retrieval info: PRIVATE: ADDRESSSTALL_A NUMERIC "0"

// Retrieval info: PRIVATE: AclrAddr NUMERIC "0"

// Retrieval info: PRIVATE: AclrByte NUMERIC "0"

// Retrieval info: PRIVATE: AclrData NUMERIC "0"

// Retrieval info: PRIVATE: AclrOutput NUMERIC "0"

// Retrieval info: PRIVATE: BYTE_ENABLE NUMERIC "0"

// Retrieval info: PRIVATE: BYTE_SIZE NUMERIC "8"

// Retrieval info: PRIVATE: BlankMemory NUMERIC "1"

// Retrieval info: PRIVATE: CLOCK_ENABLE_INPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_OUTPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: Clken NUMERIC "0"
// Retrieval info: PRIVATE: DataBusSeparated NUMERIC "1"
// Retrieval info: PRIVATE: IMPLEMENT_IN_LES NUMERIC "0"
// Retrieval info: PRIVATE: INIT_FILE_LAYOUT STRING "PORT_A"
// Retrieval info: PRIVATE: INIT_TO_SIM_X NUMERIC "0"
// Retrieval info: PRIVATE: INTENDED_DEVICE_FAMILY STRING "Cyclone V"
// Retrieval info: PRIVATE: JTAG_ENABLED NUMERIC "0"
// Retrieval info: PRIVATE: JTAG_ID STRING "NONE"
// Retrieval info: PRIVATE: MAXIMUM_DEPTH NUMERIC "0"
// Retrieval info: PRIVATE: MIFfilename STRING "spriteMAP.mif"
// Retrieval info: PRIVATE: NUMWORDS_A NUMERIC "32"
// Retrieval info: PRIVATE: RAM_BLOCK_TYPE NUMERIC "0"
// Retrieval info: PRIVATE: READ_DURING_WRITE_MODE_PORT_A NUMERIC "3"
// Retrieval info: PRIVATE: RegAddr NUMERIC "1"
// Retrieval info: PRIVATE: RegData NUMERIC "1"
// Retrieval info: PRIVATE: RegOutput NUMERIC "0"
// Retrieval info: PRIVATE: SYNTH_WRAPPER_GEN_POSTFIX STRING "0"
// Retrieval info: PRIVATE: SingleClock NUMERIC "1"
// Retrieval info: PRIVATE: UseDQRAM NUMERIC "1"
// Retrieval info: PRIVATE: WRCONTROL_ACLR_A NUMERIC "0"
// Retrieval info: PRIVATE: WidthAddr NUMERIC "5"
// Retrieval info: PRIVATE: WidthData NUMERIC "26"
// Retrieval info: PRIVATE: rden NUMERIC "0"
// Retrieval info: LIBRARY: altera_mf altera_mf.altera_mf_components.all
// Retrieval info: CONSTANT: CLOCK_ENABLE_INPUT_A STRING "BYPASS"
// Retrieval info: CONSTANT: CLOCK_ENABLE_OUTPUT_A STRING "BYPASS"
// Retrieval info: CONSTANT: INTENDED_DEVICE_FAMILY STRING "Cyclone V"
// Retrieval info: CONSTANT: LPM_HINT STRING "ENABLE_RUNTIME_MOD=NO"
// Retrieval info: CONSTANT: LPM_TYPE STRING "altsyncram"
// Retrieval info: CONSTANT: NUMWORDS_A NUMERIC "32"
// Retrieval info: CONSTANT: OPERATION_MODE STRING "SINGLE_PORT"
// Retrieval info: CONSTANT: OUTDATA_ACLR_A STRING "NONE"
// Retrieval info: CONSTANT: OUTDATA_REG_A STRING "UNREGISTERED"
// Retrieval info: CONSTANT: POWER_UP_UNINITIALIZED STRING "FALSE"
// Retrieval info: CONSTANT: READ_DURING_WRITE_MODE_PORT_A STRING
"NEW_DATA_NO_NBE_READ"
// Retrieval info: CONSTANT: WIDTHAD_A NUMERIC "5"
// Retrieval info: CONSTANT: WIDTH_A NUMERIC "26"
// Retrieval info: CONSTANT: WIDTH_BYTEENA_A NUMERIC "1"
// Retrieval info: USED_PORT: address 0 0 5 0 INPUT NODEFVAL "address[4..0]"
// Retrieval info: USED_PORT: clock 0 0 0 0 INPUT VCC "clock"
// Retrieval info: USED_PORT: data 0 0 26 0 INPUT NODEFVAL "data[25..0]"
// Retrieval info: USED_PORT: q 0 0 26 0 OUTPUT NODEFVAL "q[25..0]"
// Retrieval info: USED_PORT: wren 0 0 0 0 INPUT NODEFVAL "wren"
// Retrieval info: CONNECT: @address_a 0 0 5 0 address 0 0 5 0
// Retrieval info: CONNECT: @clock0 0 0 0 0 clock 0 0 0 0
// Retrieval info: CONNECT: @data_a 0 0 26 0 data 0 0 26 0
// Retrieval info: CONNECT: @wren_a 0 0 0 0 wren 0 0 0 0
// Retrieval info: CONNECT: q 0 0 26 0 @q_a 0 0 26 0
// Retrieval info: GEN_FILE: TYPE_NORMAL spriteRAM.v TRUE
// Retrieval info: GEN_FILE: TYPE_NORMAL spriteRAM.inc FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL spriteRAM.cmp FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL spriteRAM.bsf FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL spriteRAM_inst.v FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL spriteRAM_bb.v TRUE
// Retrieval info: LIB_FILE: altera_mf
```

tileRAM.v

```
// megafunction wizard: %RAM: 1-PORT%

// GENERATION: STANDARD

// VERSION: WM1.0

// MODULE: altsyncram


// ============================================================

// File Name: tileRAM.v

// Megafunction Name(s):

//                              altsyncram

//

// Simulation Library Files(s):

//                              altera_mf

// ============================================================

// ************************************************************

// THIS IS A WIZARD-GENERATED FILE. DO NOT EDIT THIS FILE!

//

// 21.1.0 Build 842 10/21/2021 SJ Lite Edition

// ************************************************************



//Copyright (C) 2021  Intel Corporation. All rights reserved.

//Your use of Intel Corporation's design tools, logic functions

//and other software and tools, and any partner logic

//functions, and any output files from any of the foregoing
```

```
//(including device programming or simulation files), and any
//associated documentation or information are expressly subject
//to the terms and conditions of the Intel Program License
//Subscription Agreement, the Intel Quartus Prime License Agreement,
//the Intel FPGA IP License Agreement, or other applicable license
//agreement, including, without limitation, that your use is for
//the sole purpose of programming logic devices manufactured by
//Intel and sold by Intel or its authorized distributors.  Please
//refer to the applicable agreement for further details, at
//https://fpgasoftware.intel.com/eula.


// synopsys translate_off
`timescale 1 ps / 1 ps
// synopsys translate_on
module tileRAM (
        address,
        clock,
        data,
        wren,
        q);

        input   [8:0]  address;
        input     clock;
        input   [7:0]  data;
        input     wren;
        output  [7:0]  q;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_off
```

```
`endif
	tri1	clock;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_on
`endif

	wire [7:0] sub_wire0;
	wire [7:0] q = sub_wire0[7:0];

	altsyncram	altsyncram_component (
				.address_a (address),
				.clock0 (clock),
				.data_a (data),
				.wren_a (wren),
				.q_a (sub_wire0),
				.aclr0 (1'b0),
				.aclr1 (1'b0),
				.address_b (1'b1),
				.addressstall_a (1'b0),
				.addressstall_b (1'b0),
				.byteena_a (1'b1),
				.byteena_b (1'b1),
				.clock1 (1'b1),
				.clocken0 (1'b1),
				.clocken1 (1'b1),
				.clocken2 (1'b1),
				.clocken3 (1'b1),
				.data_b (1'b1),
				.eccstatus (),
                                        .q_b (),
                                        .rden_a (1'b1),
                                        .rden_b (1'b1),
                                        .wren_b (1'b0));
          defparam
                    altsyncram_component.clock_enable_input_a = "BYPASS",
                    altsyncram_component.clock_enable_output_a = "BYPASS",
                    altsyncram_component.init_file = "mif/tile_RAM_init.mif",
                    altsyncram_component.intended_device_family = "Cyclone V",
                    altsyncram_component.lpm_hint = "ENABLE_RUNTIME_MOD=NO",
                    altsyncram_component.lpm_type = "altsyncram",
                    altsyncram_component.numwords_a = 320,
                    altsyncram_component.operation_mode = "SINGLE_PORT",
                    altsyncram_component.outdata_aclr_a = "NONE",
                    altsyncram_component.outdata_reg_a = "UNREGISTERED",
                    altsyncram_component.power_up_uninitialized = "FALSE",
                    altsyncram_component.read_during_write_mode_port_a =
"NEW_DATA_NO_NBE_READ",
                    altsyncram_component.widthad_a = 9,
                    altsyncram_component.width_a = 8,
                    altsyncram_component.width_byteena_a = 1;



endmodule


// ============================================================
// CNX file retrieval info
// ============================================================
// Retrieval info: PRIVATE: ADDRESSSTALL_A NUMERIC "0"
// Retrieval info: PRIVATE: AclrAddr NUMERIC "0"
// Retrieval info: PRIVATE: AclrByte NUMERIC "0"
// Retrieval info: PRIVATE: AclrData NUMERIC "0"
// Retrieval info: PRIVATE: AclrOutput NUMERIC "0"
// Retrieval info: PRIVATE: BYTE_ENABLE NUMERIC "0"
// Retrieval info: PRIVATE: BYTE_SIZE NUMERIC "8"
// Retrieval info: PRIVATE: BlankMemory NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_INPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_OUTPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: Clken NUMERIC "0"
// Retrieval info: PRIVATE: DataBusSeparated NUMERIC "1"
// Retrieval info: PRIVATE: IMPLEMENT_IN_LES NUMERIC "0"
// Retrieval info: PRIVATE: INIT_FILE_LAYOUT STRING "PORT_A"
// Retrieval info: PRIVATE: INIT_TO_SIM_X NUMERIC "0"
// Retrieval info: PRIVATE: INTENDED_DEVICE_FAMILY STRING "Cyclone V"
// Retrieval info: PRIVATE: JTAG_ENABLED NUMERIC "0"
// Retrieval info: PRIVATE: JTAG_ID STRING "NONE"
// Retrieval info: PRIVATE: MAXIMUM_DEPTH NUMERIC "0"
// Retrieval info: PRIVATE: MIFfilename STRING "tileMAP_RAM_Init.mif"
// Retrieval info: PRIVATE: NUMWORDS_A NUMERIC "320"
// Retrieval info: PRIVATE: RAM_BLOCK_TYPE NUMERIC "0"
// Retrieval info: PRIVATE: READ_DURING_WRITE_MODE_PORT_A NUMERIC "3"
// Retrieval info: PRIVATE: RegAddr NUMERIC "1"
// Retrieval info: PRIVATE: RegData NUMERIC "1"
// Retrieval info: PRIVATE: RegOutput NUMERIC "0"
// Retrieval info: PRIVATE: SYNTH_WRAPPER_GEN_POSTFIX STRING "0"
// Retrieval info: PRIVATE: SingleClock NUMERIC "1"
// Retrieval info: PRIVATE: UseDQRAM NUMERIC "1"
// Retrieval info: PRIVATE: WRCONTROL_ACLR_A NUMERIC "0"
// Retrieval info: PRIVATE: WidthAddr NUMERIC "9"

// Retrieval info: PRIVATE: WidthData NUMERIC "8"

// Retrieval info: PRIVATE: rden NUMERIC "0"

// Retrieval info: LIBRARY: altera_mf altera_mf.altera_mf_components.all

// Retrieval info: CONSTANT: CLOCK_ENABLE_INPUT_A STRING "BYPASS"

// Retrieval info: CONSTANT: CLOCK_ENABLE_OUTPUT_A STRING "BYPASS"

// Retrieval info: CONSTANT: INIT_FILE STRING "tileMAP_RAM_Init.mif"

// Retrieval info: CONSTANT: INTENDED_DEVICE_FAMILY STRING "Cyclone V"

// Retrieval info: CONSTANT: LPM_HINT STRING "ENABLE_RUNTIME_MOD=NO"

// Retrieval info: CONSTANT: LPM_TYPE STRING "altsyncram"

// Retrieval info: CONSTANT: NUMWORDS_A NUMERIC "320"

// Retrieval info: CONSTANT: OPERATION_MODE STRING "SINGLE_PORT"

// Retrieval info: CONSTANT: OUTDATA_ACLR_A STRING "NONE"

// Retrieval info: CONSTANT: OUTDATA_REG_A STRING "UNREGISTERED"

// Retrieval info: CONSTANT: POWER_UP_UNINITIALIZED STRING "FALSE"

// Retrieval info: CONSTANT: READ_DURING_WRITE_MODE_PORT_A STRING
"NEW_DATA_NO_NBE_READ"

// Retrieval info: CONSTANT: WIDTHAD_A NUMERIC "9"

// Retrieval info: CONSTANT: WIDTH_A NUMERIC "8"

// Retrieval info: CONSTANT: WIDTH_BYTEENA_A NUMERIC "1"

// Retrieval info: USED_PORT: address 0 0 9 0 INPUT NODEFVAL "address[8..0]"

// Retrieval info: USED_PORT: clock 0 0 0 0 INPUT VCC "clock"

// Retrieval info: USED_PORT: data 0 0 8 0 INPUT NODEFVAL "data[7..0]"

// Retrieval info: USED_PORT: q 0 0 8 0 OUTPUT NODEFVAL "q[7..0]"

// Retrieval info: USED_PORT: wren 0 0 0 0 INPUT NODEFVAL "wren"

// Retrieval info: CONNECT: @address_a 0 0 9 0 address 0 0 9 0

// Retrieval info: CONNECT: @clock0 0 0 0 0 clock 0 0 0 0

// Retrieval info: CONNECT: @data_a 0 0 8 0 data 0 0 8 0

// Retrieval info: CONNECT: @wren_a 0 0 0 0 wren 0 0 0 0
// Retrieval info: CONNECT: q 0 0 8 0 @q_a 0 0 8 0
// Retrieval info: GEN_FILE: TYPE_NORMAL tileRAM.v TRUE
// Retrieval info: GEN_FILE: TYPE_NORMAL tileRAM.inc FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL tileRAM.cmp FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL tileRAM.bsf FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL tileRAM_inst.v FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL tileRAM_bb.v TRUE
// Retrieval info: LIB_FILE: altera_mf
```

tile_RAM_init.mif

```
WIDTH=8;
DEPTH=320;

ADDRESS_RADIX=UNS;
DATA_RADIX=UNS;

CONTENT BEGIN
	0 : 76;
	1 : 76;
	2 : 76;
	3 : 60;
	4 : 76;
	5 : 76;
	6 : 60;
	7 : 60;
```

```
8 : 60;
9 : 76;
10 : 76;
11 : 76;
12 : 76;
13 : 60;
14 : 76;
15 : 76;
16 : 60;
17 : 60;
18 : 60;
19 : 76;
20 : 76;
21 : 76;
22 : 60;
23 : 60;
24 : 76;
25 : 76;
26 : 60;
27 : 76;
28 : 60;
29 : 76;
30 : 76;
31 : 76;
32 : 60;
33 : 60;
34 : 76;
35 : 76;
36 : 60;
```

```
    37 : 76;
    38 : 60;
    39 : 76;
    40 : 76;
    41 : 60;
    42 : 76;
    43 : 60;
    44 : 76;
    45 : 76;
    46 : 60;
    47 : 60;
    48 : 60;
    49 : 76;
    50 : 76;
    51 : 60;
    52 : 76;
    53 : 60;
    54 : 76;
    55 : 76;
    56 : 60;
    57 : 76;
    58 : 60;
    59 : 76;
    60 : 60;
    61 : 60;
    62 : 60;
    63 : 60;
    64 : 60;
    65 : 76;
    66 : 60;
    67 : 76;
    68 : 60;
    69 : 76;
    70 : 60;
    71 : 60;
    72 : 60;
    73 : 60;
    74 : 60;
    75 : 76;
    76 : 60;
    77 : 76;
    78 : 60;
    79 : 76;
    80 : 76;
    81 : 76;
    82 : 76;
    83 : 60;
    84 : 56;
    85 : 76;
    86 : 60;
    87 : 60;
    88 : 60;
    89 : 76;
    90 : 76;
    91 : 76;
    92 : 76;
    93 : 60;
    94 : 56;
```

```
95 : 76;
96 : 60;
97 : 60;
98 : 60;
99 : 76;
100 : 76;
101 : 76;
102 : 76;
103 : 120;
104 : 76;
105 : 76;
106 : 120;
107 : 120;
108 : 120;
109 : 76;
110 : 76;
111 : 76;
112 : 76;
113 : 120;
114 : 76;
115 : 76;
116 : 120;
117 : 120;
118 : 120;
119 : 76;
120 : 76;
121 : 76;
122 : 76;
123 : 76;
```

```
124 : 76;
125 : 76;
126 : 76;
127 : 76;
128 : 76;
129 : 76;
130 : 76;
131 : 76;
132 : 76;
133 : 76;
134 : 76;
135 : 76;
136 : 76;
137 : 76;
138 : 76;
139 : 76;
140 : 32;
141 : 20;
142 : 20;
143 : 20;
144 : 20;
145 : 20;
146 : 20;
147 : 20;
148 : 20;
149 : 20;
150 : 20;
151 : 20;
152 : 20;
```

```
153 : 20;
154 : 20;
155 : 20;
156 : 20;
157 : 20;
158 : 20;
159 : 20;
160 : 12;
161 : 72;
162 : 72;
163 : 72;
164 : 72;
165 : 4;
166 : 72;
167 : 72;
168 : 72;
169 : 52;
170 : 4;
171 : 72;
172 : 4;
173 : 4;
174 : 72;
175 : 4;
176 : 4;
177 : 72;
178 : 72;
179 : 68;
180 : 12;
181 : 72;
```

```
182 : 104;
183 : 104;
184 : 80;
185 : 4;
186 : 52;
187 : 4;
188 : 4;
189 : 72;
190 : 4;
191 : 72;
192 : 4;
193 : 4;
194 : 72;
195 : 4;
196 : 72;
197 : 4;
198 : 4;
199 : 104;
200 : 12;
201 : 72;
202 : 104;
203 : 72;
204 : 84;
205 : 4;
206 : 72;
207 : 4;
208 : 4;
209 : 24;
210 : 4;
```

```
211 : 4;
212 : 24;
213 : 72;
214 : 4;
215 : 4;
216 : 72;
217 : 52;
218 : 4;
219 : 104;
220 : 12;
221 : 80;
222 : 72;
223 : 104;
224 : 4;
225 : 4;
226 : 72;
227 : 24;
228 : 24;
229 : 4;
230 : 4;
231 : 4;
232 : 72;
233 : 24;
234 : 4;
235 : 4;
236 : 4;
237 : 68;
238 : 68;
239 : 104;
```

```
240 : 12;
241 : 80;
242 : 4;
243 : 72;
244 : 88;
245 : 4;
246 : 52;
247 : 4;
248 : 4;
249 : 24;
250 : 4;
251 : 72;
252 : 4;
253 : 4;
254 : 72;
255 : 4;
256 : 4;
257 : 4;
258 : 4;
259 : 72;
260 : 12;
261 : 80;
262 : 4;
263 : 4;
264 : 72;
265 : 4;
266 : 72;
267 : 104;
268 : 104;
```

```
269 : 72;
270 : 4;
271 : 72;
272 : 4;
273 : 4;
274 : 72;
275 : 4;
276 : 4;
277 : 88;
278 : 88;
279 : 72;
280 : 12;
281 : 68;
282 : 4;
283 : 4;
284 : 72;
285 : 4;
286 : 72;
287 : 72;
288 : 72;
289 : 52;
290 : 4;
291 : 72;
292 : 4;
293 : 4;
294 : 72;
295 : 4;
296 : 68;
297 : 72;
```

```
	298 : 72;

	299 : 52;

	300 : 44;

	301 : 124;

	302 : 124;

	303 : 124;

	304 : 124;

	305 : 124;

	306 : 124;

	307 : 124;

	308 : 124;

	309 : 124;

	310 : 124;

	311 : 124;

	312 : 124;

	313 : 124;

	314 : 124;

	315 : 124;

	316 : 124;

	317 : 124;

	318 : 124;

	319 : 124;

END;
```

rr_audio_fire.hex

```
:020000040000FA
```

```
:2000000090BD74465154B39F4D4F4C8EBF674A4D5EB8964753468EBB5D4C4C5FBB974E4F97

:200020004C94C074475150ADA44C51477DC28C49514990BF71494F52ABA94F504873C09138

:200040004A504A87C37D494F4D97BE6B494F509DB9634B4E4F9EBC674A4D58AFB15B4B4EC3

:2000600053A4B8634B4C5BB0B05B4C4D52A2BA654A504C93C17348514986C2834752477566

:20008000C0944A524768B9A44F50495DACB77445544685C08846544575B9AC6147534BA1FC

:2000A000BA734555456EBC9C4F4D4F4C9BBC744455446BB7A5534C504A93BD974B51495F64

:2000C000AEB67242574371B5B1674555457CBD994D4F4D4F97BD8E48524A56A1BC824653B3

:2000E0004A569FBC8047524C519ABD8F49524B55A3BC8647514E4C8EBE99514C52498AB8B8

:20010000A95F47564482B6AF6F4458446BACB88848524E4D8DBD9A534B554678B7AD69478B

:20012000554B5497BE904F4D544681B4B277445749579CB9A15B4858456DAAB9874A51529D

:20014000496FB4B07048544F4D90B7AB6A47564D508FB9A96B45584B538EBBA56945594A6F

:200160005390BAA66846574D5194B6AA624C5154487EB0B57C4B51554778AEB39550505411

:200180004D5DA9AFAB604E52534B7EB2B18347564F5159AAB0A252544F564A7AB2B08946DE

:2001A0005A4C5750A5ABB36B4C5550515EADACA953554F564B68B0ACA655544F574A7EB217

:2001C000AD9A4A594C5A487DB2AD9E4D574E584A74B1ACA3535452535356A5ABB1724B58BA

:2001E0004F574F9FADB0894A5A4D5A497CB1AAA65B5254535456A4ABB18D4A5A4E5A4C6AE7

:20020000ACACA9634E594E5B497FAEAD9D53545551584C8DAFAE924E57535455519BADAE15

:200220009A52555552594C90AEAC9F5952584F5B4A83AFAAA9654E5B4F5C4B74AFA7AF6E41

:200240004E58535556539AACAC995256555458506EB1A5B1714E595357545593B2A3AF6448

:2002600053565655556569EACA8A55C55565656555BA5A9AAA35A56565755585497ADA7A6FA

:200280005E55565755585285B2A1B1755258565658536BB0A2B090535955585559548EB06A

:2002A000A2B0705259565857565EA6A7AA9F5A57575856595373AEA2AD9C5A565857585904

:2002C000546EADA3AC9B59585759565A5465A6A8A5AB6A545A575958595793AEA0B0835326

:2002E0005B565A565B5370A9A6A6A86D535C565B565C5485ADA2AA9D5E575B575B575C542B

:200300008EACA2AA9B5E585B585B575D538AACA2A8A164565D575C575D5472A8A4A5A66D88

:20032000555D585C585B595A93ACA0AB995F575D575E565F5373A5A6A2AA8E575C5B5A5CA2

:20034000585E556EA4A6A0AC7E565C5B5A5D585F56689EA89EAC8A5A5B5D5A5D5A5D5A5E85

:2003600098A7A2A4A36A575E5A5C5C5B5D5968A2A3A3A49E625A5D5C5C5C5B5E5875A7A186

:20038000A4A39B5C5E5B5D5B5E5B5F5871A89FA5A0A4685A5F5B5E5B5F5B605990A5A1A17E
```

```
:2003A000A68557615A605B5F5B605A67A0A2A2A0A578566359625A615A62596AA0A2A1A098

:2003C000A47557625B615C605D5F5F5E95A3A19FA58A5A615E5F5F5E605C62597DA69DA3A4

:2003E0009DA5755A625D605F5F605E615B7AA69BA39CA6785A625E615F605F60615D73A5AE

:200400009BA29CA4915E615F6160615F615F625C84A59AA39AA5825D626061616161616125

:20042000616090A39AA19AA4815E63606261626162616260635D74A29BA09BA19462626162629B

:20044000626262626262618EA299A099A28B60636263636263626361655E83A2999F9A9F91

:2004600096676165626462656265626562656168969E999E99A08E64636562656266626626757

:20048000616B969E989E989F906663656365646564656564666186 9E999A9B989E826067D6

:2004A0006466656665666566656762889E979A99989C856069646865676567656765676510

:2004C0006D9798999899979C79626A64696569656965696569647098979897999959C7F635B

:2004E0006A66696768686769676967 6A6478999598969895 9A79656B676A686A686969698D

:2005000069696A686F9595969596959790 6A6A696A6A6A6A6A6A6A6A6B696C687B989296B1

:2005200093969397 8D6B6B6B6B6B6B6B6B6B6B6B6C6B6C6B8A96919592959197826A6D3B

:200540006B6D6C6D6C6C6D6C6D6C6D6C6F6A7C95909391929291947 96B6F6D6F6D6E6E6EBA

:200560006E6E6F6E706E716D7C908E8E8E8D8E8C8F7D7174737474747575757676767 7773A

:20058000777877808885868585858485817B7C7C7C7C7D7D7D7D7D7E7E7E7E7E7E7F7F807B

:2005A00080808080807F807F807F807B737474747474747574757475747 6747A8A898A89D3

:2005C000898989888A81757876777778777878787879797979797A79818684858484836A

:2005E000838282807F807F807F807F807F807F807F807F807F807F807F807F807F80807F02

:20060000807F807F807F807F807F80807F80807F807F7F807F807F807F807F807F807F80E9

:200620007F807F807F80807F80807F807F80807F807F807F80808080807F807F807F807FC7

:00000001FF
```

rowRAM2.v

```
// megafunction wizard: %RAM: 2-PORT%
// GENERATION: STANDARD
// VERSION: WM1.0
// MODULE: altsyncram

// ============================================================
// File Name: rowRAM2.v
// Megafunction Name(s):
//                      altsyncram
//
// Simulation Library Files(s):
//                      altera_mf
// ============================================================
// ************************************************************
// THIS IS A WIZARD-GENERATED FILE. DO NOT EDIT THIS FILE!
//
// 21.1.0 Build 842 10/21/2021 SJ Lite Edition
// ************************************************************


//Copyright (C) 2021  Intel Corporation. All rights reserved.
//Your use of Intel Corporation's design tools, logic functions
//and other software and tools, and any partner logic
//functions, and any output files from any of the foregoing
//(including device programming or simulation files), and any
//associated documentation or information are expressly subject
//to the terms and conditions of the Intel Program License
//Subscription Agreement, the Intel Quartus Prime License Agreement,
//the Intel FPGA IP License Agreement, or other applicable license
//agreement, including, without limitation, that your use is for
```

```
//the sole purpose of programming logic devices manufactured by
//Intel and sold by Intel or its authorized distributors.  Please
//refer to the applicable agreement for further details, at
//https://fpgasoftware.intel.com/eula.


// synopsys translate_off
`timescale 1 ps / 1 ps
// synopsys translate_on
module rowRAM2 (
	address_a,
	address_b,
	clock,
	data_a,
	data_b,
	wren_a,
	wren_b,
	q_a,
	q_b);

	input	[8:0]  address_a;
	input	[8:0]  address_b;
	input	  clock;
	input	[7:0]  data_a;
	input	[7:0]  data_b;
	input	  wren_a;
	input	  wren_b;
	output	[7:0]  q_a;
	output	[7:0]  q_b;
```

```
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_off
`endif
	tri1	  clock;
	tri0	  wren_a;
	tri0	  wren_b;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_on
`endif

	wire [7:0] sub_wire0;
	wire [7:0] sub_wire1;
	wire [7:0] q_a = sub_wire0[7:0];
	wire [7:0] q_b = sub_wire1[7:0];

	altsyncram	altsyncram_component (
				.address_a (address_a),
				.address_b (address_b),
				.clock0 (clock),
				.data_a (data_a),
				.data_b (data_b),
				.wren_a (wren_a),
				.wren_b (wren_b),
				.q_a (sub_wire0),
				.q_b (sub_wire1),
				.aclr0 (1'b0),
				.aclr1 (1'b0),
				.addressstall_a (1'b0),
				.addressstall_b (1'b0),
                                .byteena_a (1'b1),
                                .byteena_b (1'b1),
                                .clock1 (1'b1),
                                .clocken0 (1'b1),
                                .clocken1 (1'b1),
                                .clocken2 (1'b1),
                                .clocken3 (1'b1),
                                .eccstatus (),
                                .rden_a (1'b1),
                                .rden_b (1'b1));
        defparam
                altsyncram_component.address_reg_b = "CLOCK0",
                altsyncram_component.clock_enable_input_a = "BYPASS",
                altsyncram_component.clock_enable_input_b = "BYPASS",
                altsyncram_component.clock_enable_output_a = "BYPASS",
                altsyncram_component.clock_enable_output_b = "BYPASS",
                altsyncram_component.indata_reg_b = "CLOCK0",
                altsyncram_component.intended_device_family = "Cyclone V",
                altsyncram_component.lpm_type = "altsyncram",
                altsyncram_component.numwords_a = 512,
                altsyncram_component.numwords_b = 512,
                altsyncram_component.operation_mode = "BIDIR_DUAL_PORT",
                altsyncram_component.outdata_aclr_a = "NONE",
                altsyncram_component.outdata_aclr_b = "NONE",
                altsyncram_component.outdata_reg_a = "UNREGISTERED",
                altsyncram_component.outdata_reg_b = "UNREGISTERED",
                altsyncram_component.power_up_uninitialized = "FALSE",
                altsyncram_component.read_during_write_mode_mixed_ports = "DONT_CARE",
                altsyncram_component.read_during_write_mode_port_a =
"NEW_DATA_NO_NBE_READ",

                altsyncram_component.read_during_write_mode_port_b =
"NEW_DATA_NO_NBE_READ",

                altsyncram_component.widthad_a = 9,

                altsyncram_component.widthad_b = 9,

                altsyncram_component.width_a = 8,

                altsyncram_component.width_b = 8,

                altsyncram_component.width_byteena_a = 1,

                altsyncram_component.width_byteena_b = 1,

                altsyncram_component.wrcontrol_wraddress_reg_b = "CLOCK0";



endmodule


// ============================================================

// CNX file retrieval info

// ============================================================

// Retrieval info: PRIVATE: ADDRESSSTALL_A NUMERIC "0"

// Retrieval info: PRIVATE: ADDRESSSTALL_B NUMERIC "0"

// Retrieval info: PRIVATE: BYTEENA_ACLR_A NUMERIC "0"

// Retrieval info: PRIVATE: BYTEENA_ACLR_B NUMERIC "0"

// Retrieval info: PRIVATE: BYTE_ENABLE_A NUMERIC "0"

// Retrieval info: PRIVATE: BYTE_ENABLE_B NUMERIC "0"

// Retrieval info: PRIVATE: BYTE_SIZE NUMERIC "8"

// Retrieval info: PRIVATE: BlankMemory NUMERIC "1"

// Retrieval info: PRIVATE: CLOCK_ENABLE_INPUT_A NUMERIC "0"

// Retrieval info: PRIVATE: CLOCK_ENABLE_INPUT_B NUMERIC "0"

// Retrieval info: PRIVATE: CLOCK_ENABLE_OUTPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_OUTPUT_B NUMERIC "0"

// Retrieval info: PRIVATE: CLRdata NUMERIC "0"

// Retrieval info: PRIVATE: CLRq NUMERIC "0"

// Retrieval info: PRIVATE: CLRrdaddress NUMERIC "0"

// Retrieval info: PRIVATE: CLRrren NUMERIC "0"

// Retrieval info: PRIVATE: CLRwraddress NUMERIC "0"

// Retrieval info: PRIVATE: CLRwren NUMERIC "0"

// Retrieval info: PRIVATE: Clock NUMERIC "0"

// Retrieval info: PRIVATE: Clock_A NUMERIC "0"

// Retrieval info: PRIVATE: Clock_B NUMERIC "0"

// Retrieval info: PRIVATE: IMPLEMENT_IN_LES NUMERIC "0"

// Retrieval info: PRIVATE: INDATA_ACLR_B NUMERIC "0"

// Retrieval info: PRIVATE: INDATA_REG_B NUMERIC "1"

// Retrieval info: PRIVATE: INIT_FILE_LAYOUT STRING "PORT_A"

// Retrieval info: PRIVATE: INIT_TO_SIM_X NUMERIC "0"

// Retrieval info: PRIVATE: INTENDED_DEVICE_FAMILY STRING "Cyclone V"

// Retrieval info: PRIVATE: JTAG_ENABLED NUMERIC "0"

// Retrieval info: PRIVATE: JTAG_ID STRING "NONE"

// Retrieval info: PRIVATE: MAXIMUM_DEPTH NUMERIC "0"

// Retrieval info: PRIVATE: MEMSIZE NUMERIC "4096"

// Retrieval info: PRIVATE: MEM_IN_BITS NUMERIC "0"

// Retrieval info: PRIVATE: MIFfilename STRING ""

// Retrieval info: PRIVATE: OPERATION_MODE NUMERIC "3"

// Retrieval info: PRIVATE: OUTDATA_ACLR_B NUMERIC "0"

// Retrieval info: PRIVATE: OUTDATA_REG_B NUMERIC "0"

// Retrieval info: PRIVATE: RAM_BLOCK_TYPE NUMERIC "0"

// Retrieval info: PRIVATE: READ_DURING_WRITE_MODE_MIXED_PORTS NUMERIC "2"

// Retrieval info: PRIVATE: READ_DURING_WRITE_MODE_PORT_A NUMERIC "3"

// Retrieval info: PRIVATE: READ_DURING_WRITE_MODE_PORT_B NUMERIC "3"
// Retrieval info: PRIVATE: REGdata NUMERIC "1"
// Retrieval info: PRIVATE: REGq NUMERIC "0"
// Retrieval info: PRIVATE: REGrdaddress NUMERIC "0"
// Retrieval info: PRIVATE: REGrren NUMERIC "0"
// Retrieval info: PRIVATE: REGwraddress NUMERIC "1"
// Retrieval info: PRIVATE: REGwren NUMERIC "1"
// Retrieval info: PRIVATE: SYNTH_WRAPPER_GEN_POSTFIX STRING "0"
// Retrieval info: PRIVATE: USE_DIFF_CLKEN NUMERIC "0"
// Retrieval info: PRIVATE: UseDPRAM NUMERIC "1"
// Retrieval info: PRIVATE: VarWidth NUMERIC "0"
// Retrieval info: PRIVATE: WIDTH_READ_A NUMERIC "8"
// Retrieval info: PRIVATE: WIDTH_READ_B NUMERIC "8"
// Retrieval info: PRIVATE: WIDTH_WRITE_A NUMERIC "8"
// Retrieval info: PRIVATE: WIDTH_WRITE_B NUMERIC "8"
// Retrieval info: PRIVATE: WRADDR_ACLR_B NUMERIC "0"
// Retrieval info: PRIVATE: WRADDR_REG_B NUMERIC "1"
// Retrieval info: PRIVATE: WRCTRL_ACLR_B NUMERIC "0"
// Retrieval info: PRIVATE: enable NUMERIC "0"
// Retrieval info: PRIVATE: rden NUMERIC "0"
// Retrieval info: LIBRARY: altera_mf altera_mf.altera_mf_components.all
// Retrieval info: CONSTANT: ADDRESS_REG_B STRING "CLOCK0"
// Retrieval info: CONSTANT: CLOCK_ENABLE_INPUT_A STRING "BYPASS"
// Retrieval info: CONSTANT: CLOCK_ENABLE_INPUT_B STRING "BYPASS"
// Retrieval info: CONSTANT: CLOCK_ENABLE_OUTPUT_A STRING "BYPASS"
// Retrieval info: CONSTANT: CLOCK_ENABLE_OUTPUT_B STRING "BYPASS"
// Retrieval info: CONSTANT: INDATA_REG_B STRING "CLOCK0"
// Retrieval info: CONSTANT: INTENDED_DEVICE_FAMILY STRING "Cyclone V"
// Retrieval info: CONSTANT: LPM_TYPE STRING "altsyncram"
// Retrieval info: CONSTANT: NUMWORDS_A NUMERIC "512"
// Retrieval info: CONSTANT: NUMWORDS_B NUMERIC "512"
// Retrieval info: CONSTANT: OPERATION_MODE STRING "BIDIR_DUAL_PORT"
// Retrieval info: CONSTANT: OUTDATA_ACLR_A STRING "NONE"
// Retrieval info: CONSTANT: OUTDATA_ACLR_B STRING "NONE"
// Retrieval info: CONSTANT: OUTDATA_REG_A STRING "UNREGISTERED"
// Retrieval info: CONSTANT: OUTDATA_REG_B STRING "UNREGISTERED"
// Retrieval info: CONSTANT: POWER_UP_UNINITIALIZED STRING "FALSE"
// Retrieval info: CONSTANT: READ_DURING_WRITE_MODE_MIXED_PORTS STRING "DONT_CARE"
// Retrieval info: CONSTANT: READ_DURING_WRITE_MODE_PORT_A STRING
"NEW_DATA_NO_NBE_READ"
// Retrieval info: CONSTANT: READ_DURING_WRITE_MODE_PORT_B STRING
"NEW_DATA_NO_NBE_READ"
// Retrieval info: CONSTANT: WIDTHAD_A NUMERIC "9"
// Retrieval info: CONSTANT: WIDTHAD_B NUMERIC "9"
// Retrieval info: CONSTANT: WIDTH_A NUMERIC "8"
// Retrieval info: CONSTANT: WIDTH_B NUMERIC "8"
// Retrieval info: CONSTANT: WIDTH_BYTEENA_A NUMERIC "1"
// Retrieval info: CONSTANT: WIDTH_BYTEENA_B NUMERIC "1"
// Retrieval info: CONSTANT: WRCONTROL_WRADDRESS_REG_B STRING "CLOCK0"
// Retrieval info: USED_PORT: address_a 0 0 9 0 INPUT NODEFVAL "address_a[8..0]"
// Retrieval info: USED_PORT: address_b 0 0 9 0 INPUT NODEFVAL "address_b[8..0]"
// Retrieval info: USED_PORT: clock 0 0 0 0 INPUT VCC "clock"
// Retrieval info: USED_PORT: data_a 0 0 8 0 INPUT NODEFVAL "data_a[7..0]"
// Retrieval info: USED_PORT: data_b 0 0 8 0 INPUT NODEFVAL "data_b[7..0]"
// Retrieval info: USED_PORT: q_a 0 0 8 0 OUTPUT NODEFVAL "q_a[7..0]"
// Retrieval info: USED_PORT: q_b 0 0 8 0 OUTPUT NODEFVAL "q_b[7..0]"
// Retrieval info: USED_PORT: wren_a 0 0 0 0 INPUT GND "wren_a"
// Retrieval info: USED_PORT: wren_b 0 0 0 0 INPUT GND "wren_b"
// Retrieval info: CONNECT: @address_a 0 0 9 0 address_a 0 0 9 0
// Retrieval info: CONNECT: @address_b 0 0 9 0 address_b 0 0 9 0

// Retrieval info: CONNECT: @clock0 0 0 0 0 clock 0 0 0 0

// Retrieval info: CONNECT: @data_a 0 0 8 0 data_a 0 0 8 0

// Retrieval info: CONNECT: @data_b 0 0 8 0 data_b 0 0 8 0

// Retrieval info: CONNECT: @wren_a 0 0 0 0 wren_a 0 0 0 0

// Retrieval info: CONNECT: @wren_b 0 0 0 0 wren_b 0 0 0 0

// Retrieval info: CONNECT: q_a 0 0 8 0 @q_a 0 0 8 0

// Retrieval info: CONNECT: q_b 0 0 8 0 @q_b 0 0 8 0

// Retrieval info: GEN_FILE: TYPE_NORMAL rowRAM2.v TRUE

// Retrieval info: GEN_FILE: TYPE_NORMAL rowRAM2.inc FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL rowRAM2.cmp FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL rowRAM2.bsf FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL rowRAM2_inst.v FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL rowRAM2_bb.v TRUE

// Retrieval info: LIB_FILE: altera_mf
```

rr_audio_fly.hex

```
:020000040000FA

:200000007C7C7C7D7C85909192929393939393919391948B807F7E7D7C7D7B7C7B7C7B7C1E

:200020007B7D7B7B7B7A7C7C7B7C7B7C7B7C7C7C7C7C7B7C7B7C7B7C7C7C7C7B7A7B7A7B52

:200040007B7A7B7A7B7B7C7B7C7B7C7C7C7C7B7C7B7C7B7C7B7C7C7D7D7D7D7D7D7C7E7D25

:200060007E7D7E80807F808081818081818081808180818080807F807F807F7F7F7F7E7F8A

:200080007E7F7F7E7F7E818382838283838283837F7E7D7E7D7E7D7D7C7D7C7D7C7D7C7D7F

:2000A0007C7D7D7D7D7D7D7E838F929090908F908E908F8F8F877D7A7576757476747476F4

:2000C000746E746B7073717270796E696870656A6865627997989E9395989A9DA090939B68
```

```
:2000E000919E95839B828D97949496867F66283D33414640443E3B434B404E4E5156495584
:200100005D544F5B574F495466635E65595F6A686D5C636B715F6C69686E6B676B6CA3E0ED
:20012000CED5CAD4CAE4E6EFEAE6F1E1DFDDE8DCE3DED5DDD7D0D5D0D3CFCAC6C2C6C4C5D7
:20014000C5B9BFBCBC9D8A8E838882835549464042 3D4041423F4343443E403A3F434544F3
:2001600047454848494B474A4D4E50534E525156535955595D5660828B8B908C918E8F9258
:20018000928D8D8C92908C8C8E8D8A8E8B8C85898789888A888A89868989888858261585B9D
:2001A0005555535755557545B5B5C5D5D60605F5F62626663666667646D67686865656C4E
:2001C000667AA9A6AFACB3ADC0E1DBE3DFDDDDDFDBDDD7D5D5D0CFCECCC8C8CAC2C1C0C08F
:2001E000BEC0BFBAB6B6B3B3B3B0AEAAAF906B726E6E6A6F65403E3D373B3B3936393839C4
:200200003A393E423D3F454345434849484A494C4B4F4F4F545055555A5877898289878C51
:200220008B8F8C8E8D8E8D898C898A888A8A898888898A8A898C888A88868487868888838F
:20024000898583838787858686868087765 65B545855575658575A5B5A5A5D605E5E5C5F3D
:200260005E5E6061678E969A9CA0A0A7CAC9CDCCCDCCCECCCBCAC6C6C2C3BFC1BBBCB9B6EE
:20028000B5B1B2B0B2AEAEAEABABA7A7A3A795716F6B6C706E706F6F6E706F71706F6D6901
:2002A000696B696D573E433B3D3F3E3E414246444344464A4A4D4B4C4D504F505066807A4B
:2002C000818184848584848788848687858685858685878487878385848784858082 81858A
:2002E0008382818482857D696F6C6C6E6D6D6E7070716E737072707573757574757377766B
:2003000077757575737877594A7A6A7A8AAAAC7D1CECFCED2CECFCCCBCBC8C6C2C3C1BD47
:20032000BABCB9B6B8B4B3B093898A868583837C5D50514D4E4D5554555A585B595D5D5D75
:200340005E5A585859585A595A5A5D5F5C5F60606263635F62626362656564666767 6A64A6
:2003600070898D909191928D9390909490908F91908E8E8C8F8B8E8C8D8E8B8C8D8C8C84D8
:200380006B6565646562656467656666 6C6967696A676867696B6D6C6E6E6D6F706E6E6F37
:2003A00070707370717472727272 6F839999A1A0A29EB4C4BFC3C3C3C1BEC2C0BCBCBBB7BD
:2003C000B5B2B2B1AEAEACA9ABA9A5A6A6A3A4A1A1877677766F726750555453595A5957F3
:2003E0005859595B5552555357565658555958585 95B5F5C5B5D5C63616364666362626194
:20040000676A66678087888D8787898F8A888D9AAAA8A8AAADA8AAACA6A4A6ACA6A9ABA935
:20042000ABA6A7A6A6A6A6A7A5A8A5A0A1A1A39585755354514D4A474947474D4C484C4E57
:200440004F4C4F52534F555453535B565B58607B7F838384828C9A9AA09D9C999A9C9C9B06
:200460009E9C99969A9B96959994959892949595939583716E6B6C6A6E6E6B6C70707474C3
```

```
:200480007573757777777764575A575A54595959595B5E5C5E5F60615F62636166646667F6

:2004A00068696B6A6C6A6B7A84838684858788888A86A0C3C2C3C2C2BEC0BBBCBBB9B8B65B

:2004C000B5B2B1B0AEADACA7A9A8A6A4A1A3A1A19D9D9D9B94756F6E6B6B6A6B686A6B5CE9

:2004E000565755575859595959 5B5B5D5D5E5F616164798182858585889999 9B9B9B9C9CC6

:200500009D9B9B999A9999979796969595939491949193 8D797371716F6F6E6E6E6F6E71DE

:200520007776777778787978797875 6F646969696A6A6B6C696C6B6C6D6D6E6C6F6F7070C2

:2005400070717172717273737573 7B86838685888688878786899 99C9E9E9E9E9C9D9B9CC7

:20056000989A958A8986868685848584838383838282818381828180746F6F6D6D6C6D6D73

:200580006C6D6D6E6F6F6F70707070727172737373747375738 48F91929294929DA09F9FB4

:2005A0009F9F9E9E9D9D9A9C999A999898979796949493949181777874766E6868676866E4

:2005C0006867696F6E70707271737273726F6C6E6C6D6D6F6F706F70707170727272727133

:2005E000747473747574757476757677767980818183828383848384828A8D8C8D8D8C8C24

:200600008C8B8C8A8B8A8B8A8989888A888887898687827C7E7C7C7B7C7B7C77747374738B

:2006200074747375747575757676767776777877787777979797879777828F8F929293919622

:200640009A99999999989898979795959495939595959495939 6887E7D7D7C7D7B7B7B7CC9

:200660007B7C7C7B7E7D838687888481818185837F7D7E80817F838587868687868C745E6A

:20068000645E5C5F5A5C555959555B5957585856565357595557 5E626064635E656C6E6EC3

:2006A0006B706E6C726F6E737376787472 6E686B67676C6B6E696D6A6D6E6F6E73756E6B6F

:2006C0006D726D6B6D6D717175716E7372716E737174757270747577797675 7B878E8D8D72

:2006E0008C918E92919094929492958F8E9CB2AEB5B0DCFCF3F7F3F3E9E8E4E5E2E3D1B2B3

:20070000AFAFA9A6A4A1A1A19D9C9E9F9C9B9A9C999996949293928E8E8F8A888A858971E9

:200720004B4941413F423F3F40434543434446474A494E4D4F515355565858595A5A619803

:20074000A7A3A5A2A2A1A1A0A09E9D9C9B9B999A7564635E5F5C5E5B5D5D5F5D606161629C

:200760006264656566666869686A6A6B6B6D6C6C6C6B6E6E6F6F707170717173737 3747EE6

:20078000828384838584868587858686858686869 1ADABB1ADBEE5E7E8DEDCDAD9D6D4D268

:2007A000D1C8ABA7A4A19F9C9C9A999798989997989697969694929 0908F8E8E8D8F8C8FEF

:2007C000835C555250514F514F515151535355555758585 95A5B5B5D5D5F5E6060626163F3

:2007E000646474818084838585858686868486848583867 86B6B696869686 9686A696B6A0F

:200800006C6B6D6C6E6D6F6E6F706F717071727173727373747575757675767677767 8779C
```

```
:2008200078777F848686878788878887888788878888888997989A999FC0C5BFC1C0C0BE0B
:20084000BDBCBABAB7ABA5A4A1A09E9E9C9C9A9A9B9C9A9A9A989997959492929291918FBA
:20086000908C6D5C5D595A5759575A585B5A5C5C5C5E5F5E606061626364636564666673D
:200880006868696A696F777679787A7A797B7A7B7B7A7B7A7B7C7870716F706F70707170D9
:2008A000717172717372737374747475747676757776777678777878787978797877A7A7982
:2008C0007A7A7A7A7A7E8787898889898989898A888988898988888D8F8F908B9DAFADAFB9
:2008E000AFAFAFAFACADACAAA6A2A1A09E9E9C9C9A9B9A9A9B9A9A98999798959492939119
:20090000928F927B656561616060606060606261636264636564666667686769696A6B6AB2
:200920006C6C6B6D6C6E6D6E707374737574757576767676777677767877747574757557558
:2009400074757576757677767777777877787879787978797A7A797A7A7A7B7A7B7A7B7B7B8F
:200960007B7B7C7C7B7C7C7C7F868989898A8A898A898A8A8A8989888989898B88817C8DDA
:20098000A1A2A5A3A5A4A5A3A4A2A2A19E9F9D9D9C9B9B9999999999999898979797979393B8
:2009A00091929092856D69686666656665666667676768676968 6A696B6A6C6C6B6D6C6E60
:2009C0006D6F6E6F6F706F717071717274747475747676757676777677767877787676777A4
:2009E000767777877787778787879787978797A797A797A797B7B7B7A7B7A7B7A7C7B7C7B7CCA
:200A00007C7B7C7D7C7D7C7D7C7D7C7E8489888A898A89898A898A888889888898888837E65
:200A20007F7B84999E9EA09FA09FA09E9F9D9C9B9A9A989898969795969496969494959593CF
:200A40009490918F8F8B766D6D6A6B696A696A696B6B6A6C6C6B6D6C6E6E6E6F6F6F6F7032
:200A60006F7171707272717372737473747375757777767877787777978797978797A797ACD
:200A80007978797879787978797879797A797A797A7A7B7B7A7B7A7B7A7C7B7C7B7C7B7C7B7C14
:200AA0007C7D7C7D7C7D7C7D7D7C7D7D7D7D7D828989898A898A8A898A89898889888983DA
:200AC0007E7E7E7E7D80929A999C9B9C9B9C9B9A9A9998969695959593949293919392 92D3
:200AE0009191918F8F8D8E7C71706F6E6E6D6D6E6D6E6D6D6F6E6F706F70707171717 27124
:200B0000737372737374737474757475757675767776787879787A797A7A797A7B7B7B7B00
:200B20007B7B7A7B797A797A797A797B7A7A7B7A7B7B7A7C7B7C7B7C7C7C7B7C7B7D7C7C5B
:200B40007D7C7D7C7D7C7D7C7E7D7E7E7D7E7D7E7D7E80888889888A8A898A888989898856
:200B6000867E7E7C7E7F7D7F7D8C969697989798979796979695939392929291909 08F8FBA
:200B80008E908F8E8E8E8D8D847474717271707170717071707271727272737373747 37424
:200BA000737574757476757676777776777787877787777979787A7A7B7B7A7C7B7C7B7C39
```

```
:200BC0007C7C7B7C7D7C7D7A7B7B7A7B7A7B7A7C7C7B7C7B7C7B7C7C7C7B7D7C7D7D7D7CA0

:200BE0007D7D7C7D7C7D7D7D7E7D7E7D7E7E7D7E7D7E7D7F7E8086878888898988898889DC

:200C0000878986807D7C7D7C7E7B7D7B848F909192929092909191919190909090 8E908EA1

:200C20008F8D8E8C8D8C8F8B8C89797875767674767573757375757575757675767577751F

:200C400078767676767677737B7F807D7F808182828181808181 7E807F7D817F80828082E7

:200C60008181838284838285868582777375747475747476757776767677797B7A7C7D7C10

:200C80007C7D7E7C7F817C7D7D7B7D7C7D7B7A787B7977797876757475 7A7A7C7A79777A09

:200CA0007877757274706C6869666765646665 6A757A7D7E7F80828181848485868689875C

:200CC00089898F8F8F8D8F909091958A848383868586858788858685858283807F7D7D7C2B

:200CE0007A787877737570716D706A7FADAFACABA6A6A2A09D9B9898969492918F8E8E8F54

:200D00008F908F9192919493959495969994756B6D6A6A6B6D6D6F717274757678787A7BCD

:200D20007E7E7F7F818181807F7D7D7B7A7978787775747272 70706D6D6D6B6A6D71706FE2

:200D40006F6D6E6C6D6B6B6C696261605F606163646667747B7D7E818383868689898C8D46

:200D60008E8F909191939395949696989189898989888A898B8B8989878785858382 8180FB

:200D80007E7E7D7B7B787976767575727370738489898A888988878684848382817F808082

:200DA0008181838384848687878889898B8A8C8C8E8078787878797B7A7C7C7E7E808181F6

:200DC000828383858586868485838381 81807F7D7D7C7A7B7979777776747573737271737F

:200DE0007678767777767575747473716B6B6A6B6B6D6E6E71707B838588898B8A8C8C8EE6

:200E00008E908F919292929393949494959287858585848585868585848482818280807E96

:200E20007E7D7D7B7B797A787877777675757374737 2797C7D7D7B7C7B7B7B7A7A79787992

:200E40007A797B7B7C7D7D7F7E80818182828484848685878 37F7F7F7E807F81808283828D

:200E6000848485858686878785868484838282808080 7E7E7E7C7C7B7B797A79787877777A

:200E80007676767A7C7B7C7B7A7B797A797876717272737474757676777D898A8D8E8F8FE8

:200EA00090919091919192929292939293939286838282818281828382 80817F807F7D7E17

:200EC0007D7C7C7B7B7A7A797978797777777677767575747575787778787677767775761C

:200EE0007676777778797 97A7A7C7C7D7C7E7D7F7F808180828182828080818182818 3823E

:200F00008383848584858586858686848583848283828081807F7F7E7E7D7D7C7D7B7C7CA5

:200F20007A7B797A7A787D7F7F7F7E7F7E7E7E7C7E7A7677767777787879797A7D898E8E23

:200F400090909191929292919291929291919291887F817E7F7E7F7E7F7F7E7F7F7F7E7E87
```

```
:200F60007C7D7C7C7B7C7B7B7A7B7A797A797879787877787877777767877787877778784F
:200F800076777677777878797A7A797B7A7C7C7C7D7D7E7D7F7E807F808081807F807F81D0
:200FA00081808281828183838382848483838283828282818180807F807E7E7F7D7E7D7E14
:200FC0007C7D7D7D7C7C7B7C7B7D8181808180817F80807F7C797A797A7A7A7B7B7B7C7C76
:200FE000878F8E919092919292919290929091909 28B807E7E7C7D7D7C7D7C7D7D7C7E7CFB
:201000007D7C7D7B7C7C7B7C7B7B7A7B7B7B7B7A7A7A797A797A7979797879787A7A7A7985
:201020007A797A797A797A797B7A7B7A7C7C7C7C7D7C7E7D7E7E7E7F7E7F7F807F807F7F24
:201040007E7F7F807F808080818180818081818281818081818 0807F807F7F7E7F7E7F7D96
:201060007E7D7E7D7E7D7C7D7D7C7D7C7E8082828182828081808 17E7B7C7A7C7B7C7B7DB1
:201080007B7D7C84908F9191929192919291929191918E817D7D7C7B7B7B7B7C7C7B7C7B8E
:2010A0007C7C7B7C7B7C7B7C7B7C7B7C7B7B7A7B7A7B7A7B7B7A7B7A7B7B7B7A7B7A7B7AD1
:2010C0007C7C7B7C7B7C7B7C7B7C7C7C7D7D7C7D7C7E7D7E7D7E7E7F7F7F7F7F7F7F80806A
:2010E000807E7F7E7F7E7F7F7F7E7F7E807F807F80807F80807F807F807E7F7E7F7F7F7E0F
:201100007F7E7F7E7E7D7E7D7E7D7E7D7E7D7D7E8081828182818282 81827E7D7D7C7D7DF7
:201120007C7D7D7D7E7C838E9191929293939292929291918 47E7E7B7C7A7B7A7B7A7B7A0B
:201140007B7A7C7B7C7B7C7C7C7B7C7C7C7C7C7B7C7B7C7B7C7B7C7C7C7C7C7C7C7B7C7B1B
:201160007C7C7B7D7E7E7E7E7D7E7D7E7D7E7D7E7D7E7E7E7E7E7F7E7F7F7E7F7E7F7F7FB5
:201180007E807F807F7D7E7D7E7D7E7D7E7D7E7F7E7F7F7E7F7E7F7D7E7D7E7D7E7C7E7D8F
:2011A0007D7D7C7D7B7C7B7B7B7C7C7C7C7B7D7C7C7C7A7D7F7F7E7E7F7C7D7979767673BE
:2011C000757574756F737374707381858282848584828084868 17F67676463607EC4C8C925
:2011E000CCC9C5C4C4C5C0BCBCBCBEB6B3B2AEAEB0AEB0ADA8A8A3A5A3A2ABABA8A8AAAB8C
:20120000A8ADAAA8A8A7A2A5A1A5955542403C3A3C3F3C3C3A3B3D3C3F403F444544434575
:201220004C494C504D51525153545855555A575958595A5A5C5E5F5F5D5D616061636260AB
:201240006463656468646766686969849799 9CA0A09CA09EA09B9EA1A0A1A0A4856A6C6601
:201260006561605F626362636262636363 6B716D717072727070717271716968676C677341
:20128000A3B1B2B4B6B9B7B6B5B4B3B4B1B2B0B1B1AFADABABA9AAA7A9A4A7A2AFDCE3E2E1
:2012A000E1DFDDDAD9D5D6D2D0CECBC89F7E59474641403E3F3E3F40404242454547494 94B
:2012C0004C4C4D4E5051535454565757595A5A5C5B5D5D5F5F61626263636565666 76867AE
:2012E0006969696B6A6C6C6D6D6E6E6F706F8196979A999B9B9A9B999A989B9999989A8B61
```

```
:201300007673716F6E6E6E6C6E6D6E6D6F6E707075797B7C7C7D7D7E7F7E7F7E7C7675752C
:201320007476759DB2B1B6B5B8B6B7B6B5B5B3B3B1B1AFAFAEACACAAA9A9A6A7A4A5A3A39A
:20134000B4BAB9BAB8B9B7B5B6B3B4B0B29775726E5D5251504F4F4F504F51515354555595
:20136000575759585B5B5B5D5E5E5F5F6062616363656466666767696 86A696B6A6C6B6D1D
:201380006D6D6E6E6F6F7070717172717273737472798485878788888988898888889888AF1
:2013A000888A867B787876767576757675757676767676797F8181828382838284838380A0
:2013C0007A7A79797B7791ACADB0B1B1B2B1B1B0B0AFADADABABABA8A8A6A7A4A5A4A2A2ED
:2013E000A0A09EA4A8A8A6A7A5A6A3A5A1A496746D6A6867615C5A5B595B5A5C5C5D5D5DF7
:201400005F605F61616262646364666567666868696 96A6A6A6C6B6D6D6D6E6E6F6F706FCE
:2014200071707171727372737374737475747575767577776787D7D7F7F7E807F80807F80BA
:20144000808180818181 7C7B7A7A797A797A7A797A797B7A7B7B7C828485858686868586BD
:201460008586817C7C7C7B7C7A87A5A8ABABADACADACACAAAAA9A8A7A6A6A5A3A3A1A2A00D
:20148000A09F9E9D9D9C9B9E9C9C9C9A9B999A95796A6B66676565646361636362646 3652E
:2014A0006566656766686769696A696B6A6C6B6D6D6D6E6E6F6E707070717072727 17373A3
:2014C00072747473747475757576757677767776787778777878797 87A7A7B7B7B7C7B7B27
:2014E0007C7B7C7D7C7D7D7E7D7D7B7B7C7B7C7C7B7C7B7C7B7D7D7C7D7D82858685878631
:2015000087868786868 27E7E7E7D7D7C809CA5A5A7A7A7A7A6A6A5A5A4A3A2A2A0A09F9F18
:201520009D9D9B9C9B999989898979897959795958 16D6E6A6A696868696968686767686847
:2015400069696A6A6B6B6B6C6C6D6D6E6E6F6F6F706F71707172717272737273747374 74AA
:20156000757574767576757776777677787778787779787979797A7A797A797B7B7C7D7C69
:201580007D7C7D7C7D7C7E7E7D7E7D7E7E7C7D7B7D7D7C7D7C7D7D7C7D7C7D7D7D7F85869D
:2015A00085868587858785868 07E7D7E7D7D7E7C929E9FA0A1A1A1A1A0A09F9F9D9E9C9B2D
:2015C0009C9A9A99999897979696949594949495929588 74706F6D6D6C6D6C6D6C6D6D6B5E
:2015E0006C6C6D6D6E6D6E6E6F706F70707170727272727372737374737575747575 7675C6
:20160000767577767777777877787779787978797 87A797A7A797A797B7B7A7B7A7B7B7DB1
:201620007C7D7D7E7D7E7D7E7D7E7D7E7F7E7F7E7C7D7C7D7D7C7D7C7D7D7D7D7D7D7D7E03
:201640007E8485868586868586858 6817D7F7E7D7D7D7C889A999C9B9D9D9B9C9B9B9A991C
:20166000999898989696959594949392939192919192918E7974727270707070716F71708B
:201680007170706F7170717072717271737273747474737574757476757675767677677E3
```

```
:2016A00076787778777779787978797A7A7A797A7A7A7A7B7A7B7A7B7A7B7B7C7C7B7CF8

:2016C0007B7C7E7D7E7D7F7E7F7E7F7E7F7E7F7F7F7F7E7E7C7E7C7E7C7E7D7E7D7E7E7D52

:2016E0007E7D7E7D8385858685868686868581 7D7F7D7E7D7D7D82949597979897989797 02

:201700009696969595949493939292919190908F8F8F8E8E8E8176777474747373737374C8

:2017200073747475737372747374737474757474767576767676777677777777677777806

:2017400078777777777977797C7F828181838386848586868B89888D898C8B8D898E8D773B

:2017600074767B73736D6C766C72726B7168656F6D6B6C5F67675C645A6A68576A5F617217

:20178000645D6E756E78706771686F7673786E6D6A7171797A7368767285B6C8C3C5C5BB57

:2017A000CBBCC1B7B8C0B9BFB6BABCBAADADB7B2AAAAA8B69F9F9968605F5A5757565759D3

:2017C0005F5A575A5A5E5C6361605A56665E5A5E605D5F635F6C66656764696667666869EE

:2017E000656A6C686C6B69636C716B89A0B5CCCDD0CCD0D2CBD7D4CDC9CCC8C4C6C8CAC426

:20180000C1C1BFBFBBBCBFBBB8B8B3B4B7B1A96C4C473D3D3836393B3D3F403F4142404492

:201820003F46464846454D4D4C4B5352535D59575F5E5C585A585C5E5C6060646993A59D39

:20184000A5A6A6A4A9A7AAA6A2A8A6A2A8A09DA39EA29B9AA197A09889858B868383848328

:2018600082878386818 46F5257555550534F51505055585D5C5F575A5C5F5E5E605D615F42

:201880006161666569686 76A67696B686C65809BB1C9C5C8CCCCCAC9CDC8C6C5C2C1C1BE36

:2018A000BDBDBBB6B9B6B8B3AFB0AFAEAAACAAAAA4A7967F7B797775767374767176 5D475F

:2018C000474847484747494 84C494D4E4D4D5359585A5C5B595C5D5C5A5F5E5D5F61625F89

:2018E000849C9BA1A0A2A2A1A5A2A3A2A3A5A2A09F9F9F9D9C9C9B9A958D868784838285CD

:20190000818 3817E81858281817F8380817F818082818080808363535852555454545555 56

:2019200057595A5A5B5B5D5D5C605F615F616166626D8D98A5A6A9ABA9AAA7A9A8A8A5A500

:20194000A2A3A4A2A0A09E9D9C9E9B9C9797979999959493969184808 27F7E807D7E7E7D3D

:201960007E797D7C7B7A7D797C7B7E7C7B7A5B5257565A5658575C595D5B565959656A6F3A

:201980006C746D8BAAA8AFAFAFAEAFAFB0ACAEABA9ACA8A8A7A6A6A6A49A746A6A66656060

:2019A000636464646566668646664666868696A6A6A6C6D6B6D6C6A6E6E6E6E6F6F6E6F6FF1

:2019C0007271726E71707272737273727573747675747377779 49EA2A3A4A7A5A4A5A3A5A1

:2019E000A3A1A19EA09D9E9A9E9A9B98979798959A8770746D706D706C6A6667676767672B

:201A000069676968686A6B676B6B6A6D6E6C6D6D6E6D7170717473737376747675737374F7

:201A200073727575777580A4AAABABADADADABA9ACAAA9A9A8A6A6A5A5A3A49C8F8E8C8D6D
```

```
:201A4000898A878888878886887362676465616362646562666566656666768676A686897
:201A60006A6C6A6D6B6D6D6E706F726F71717170727071757474757386959496969896974B
:201A8000969897939593949394939392928F908E8F8D8E8E8D8F8D8C8C8C8B8A888886895B
:201AA0008772686A68686768686868686868696B696B6A6D6D6E6C6F717371727272737386
:201AC0007172727272737273727595A1A1A5A3A7A6A6A4A5A3A3A2A1A0A0A09E9B90898AEE
:201AE00086888686838585828383828382828384828181818181807F806B66686566636654
:201B00006666676668686A696A6A6C6B6B6B6D6C6E6E6F6E6F706E7170736E7B89898D8EBF
:201B20008F8E8E8E8E8F8E8D8D8D8B8C8B8B8A8B8C8B8B8A8B89878B878889878988888648
:201B4000878685848285838584838284828380827E6E6C6C6A6B6B6B686B6B6E6F70707171
:201B60007271717071706F7272716F73718599989C9B9E9E9D9D9D9B9C999A9A989A8E8461
:201B8000848182818080807F807E7F7F7F81808081817F7E807F807F817E7F7E817E7E7F4D
:201BA0007E7E7F7F7D7F7E80796A6B6A6C6B6A6B6B6B6D6A6C6E6C6E706D706F71707481DA
:201BC0008386888A8F9091918F90918F908F8F8D8E8F8D8E837B7C7B7A7B797A7A7A797A33
:201BE000797A7A79777877797879787A7879797A787B797B7B7A7A7A7A7C7B7B7B7A7C7CB0
:201C00007D7D7E7E7E7D7B7D7D7C7C7C7C7B7D7B859C9DA0A1A0A1A09F9FA09E9E9B9D9534
:201C200088888584838381828181808177737471737273727372747274747374737574767A
:201C4000757577757777767878767877787878787979787A797A78797A7A797A7A7B7A7B7C
:201C60007A7B8388878989898A8989898A8989898788878788868787868685868685858593
:201C8000858585847B7A787876777678767677767876787778787978787979797A79797912
:201CA000797B7B7B7C7B7C7B7C7B7B7B7B7B7B7C7B7B7B7C7E9199999B9A9A9A9A999A999957
:201CC000978B83828180807F7F7F7E7E7D7E7D7E7D7E7E7D7E7C7E7E7D7E7D7D7676757635
:201CE00075767777787779797A797B7A7A7C7C7D7D7F7F8182818283817877767477747694
:201D000073767370707376747778777877757676757271787477726F6E7070696F706A6C6C
:201D20006A686966636769635F64636468605C625F5C5D695E63686664606368666C6E6617
:201D40007870767878807C8380787B7C7B81777D7873716A6C716F7982848B87898A898B0D
:201D60008A858B817C7673747B757773A2E1D9DCD8D6D1D0CDCECAC4C1C1BFBCBABABAB901
:201DB8000B8B5B3AEB3AEADACACACA9A4A3A0A19EA3C5CECBCECCCBCCCBC49481817A7E78CD
:201DA000787551413F3D3D3D403D43414345454545454849494D4D4E4F51515251545655BC
:201DC000555656565859595B5C5D5D5D5F62626262636464656868686A696B6C6B6E716FC2
```

```
:201DE000736C8DB8B2B5B3B4B1B0AFAEACABABA9A9A5A894746F6D6B6A686B7374767476BF
:201E0000767777777777878716F706F716F726E7BA6B0B3B5B5B6B5B6B6B5B3B3B2B0AFAE62
:201E2000ADAEAAACA9A9A7A7A5A5A3A4A1A2A19F9F9D9E9EB5C2C1C2C2BFC19E8080797AF8
:201E400076757573745E4F504C4E4D4D4E4E504F5252535455565658585A5A5C5C5D5E5F4D
:201E60005F6161626364646565676668696866A696B6A6C6D6D6D6E6E6F6F7070717172712A
:201E800072737374737A8C909092919391939192909290918F8F8C7A7675737472747A7CCB
:201EA0007C7D7D7E7E7E7F7E7F7D787576757576757678 9DAFADB2B1B3B1B3B1B1B0AFAE31
:201EC000ADACABAAAAA8A8A6A6A4A4A2A2A1A0A09F9D9D9C9B9B99A2A9AAA9AA9775726E29
:201EE0006C6C6B696A69696A645A5B5A595A5A5C5B5D5C5E5F5F6060626163646365666 6A6
:201F000066676769686A6A6A6B6B6C6D6C6E6D6F6F6E706F717071717273737274737475 00
:201F2000747575767675777 6787F848385838584858485848584858583847C7978797879E3
:201F4000787E828183828384828482848 17C7A7A797A79797A7992A9AAADADAFADAEACAD05
:201F6000ABACA9AAA8A8A7A5A5A3A3A1A2A0A09E9F9C9D9C9C9A99999997989C9D96746CBC
:201F80006A68676666666666667666866636163626463646466656767686869686A696B6A75
:201FA0006C6C6C6D6E6D6E6E6F706F7071707271727273737274737474757475757676576F8
:201FC0007677767877787877797879787B7C7D7C7D7C7D7E7D7E7E7D7E7D7E7D7E7D7C7AA5
:201FE0007C7B7C7A7F8484858486858684858684 7E7C7C7B7C7C7B7D7988A3A5A8A7A9A8F6
:20200000A9A8A8ADACACABABA9AAA9A7A8A7A7A6A4A4A3A2A0A29F9F9E9F9E9E9C9C9A8055
:2020200070726D6F6A625F605D605F605F6060626261626363646466656665666 76A696908
:202040006B6B696A696C6C6C6E6D6D6C6E6D6E6C6E70706F7072707070727170737171 73B7
:202060007072727373747375747673737475757776757476757577747674757775757477D0
:20208000757776787776797 87D80807E7E80817F7E7F7D7776777675787574767B979C9FA2
:2020A0009EA0A0A19EA9D9E3E0E0DDDCD9D9D5D3D0CECDCBC8C7C4C2C1BEBEBBBAB7B6B6AB
:2020C000A48F8C8987858484 6554514F504D504E4F4F5151535354545657585959 5A5A5DBA
:2020E0005D5D5F5F606162626364646665676868676969 6B6A6C6C6D6D6C6E6D6F6F706FF2
:2021000070707171717273747473747474757576767576777877787877797879797A797913
:202120007A7A7B7A7B7A7C7B7C7B7D818484858586868586858 37E7E7D7E7E7D7E7D7E7DBC
:20214000929C9D9E9FA09E9F9EAFB8B6B8B6B7B6B5B4B3B2B0B0AEADADAAAAA8A8A6A5A33C
:20216000A49A8681807D7D7C7B7A7A716364626262626363636465656666676768676 96864
```

```
:202180006A6A6A6B6C6B6D6C6E6E6E6F6E706F717071727172727372747374747574757540

:2021A00076757777777777778777877797879787979797A797A797B7A7B7A7B7A7B7C7B7C7B01

:2021C0007C7B7D7C7D7D7D7C7D7C7D7D7D7D7D7E8284858586868586858683 7E7F7D7E7D7E7D09

:2021E0007E7E7D8999999B9A9B9A9A9A9EA5A4A5A4A4A2A2A0A19F9F9D9E9D9C9A9B99999B

:20220000989795847C7A79787777777767676746E6C6D6C6D6C6D6D6E6D6F6E6F706F707008

:2022200071707272717372737473747475747575767576757776777678787877787879 78FE

:20224000797879797A797A797A7B7A7B7A7B7A7B7B7C7C7B7C7B7C7B7C7C7D7C7D7D7C7D21

:202260007C7D7D7E7E7D7E7D7E7D7E7D7E7D7E7F7D818485858686858685858 27F7E7F7D5E

:202280007E7D7E7E7F7E839494979596959696959997989797959695959493929191 8F9042

:2022A0008D8F8E8E807476737371706E7170706F6F6E6A6B6A6B6A67696867666A696665F3

:2022C00063686865656562626665 5E636763605E5A5F625F686B66676568666B6B6A686966

:2022E0006F696E6F6D6F757075757476767B78757471757777787178747472797186CDDA9C

:20230000D4D5D2D1CFCAC9C7C6C4C2C0BCBCB7B8B1B1ADB1AEB2B0B2B1ADADACACABAAA774

:20232000A7A6A8A4A6A3A3A4A3A2B1966D6D6868666667686A6A6C6C6F6F71707070707078

:20234000706F6F6E6E6A5F5F5E5D5C5C5A5A5859585757565756535453525352525251 5014

:202360005150515153545657 5A5B5E5F636566696B6E70717477787B7C7F80818484878812

:20238000898B8C8D8F908F9191929392919190 8E8E8B8B898786858282807F7D7B7A797626

:2023A00078A0B4B1B4B3B3B2B0AFADAAA9A6A5A2A29F9E9C9D9B9C9D9EA1A0A2A2A2A3A490

:2023C000A3A3A1A3A1A4A2A5A3A6A3A89A7A787675747575757 7787B7A7D7C7E7D7C7C7BBE

:2023E0007A7A77787576726863636 15F5F5D5D5D5B5B5A5A595959585857555655565556BC

:20240000555554565658595C5D5F60636467676A6B6E6E717274767679797C7D7E80818175

:2024200084848686888889898B8B8D8D8D8C8C8A89888785858383828080 7F7E7D7A7B79F5

:2024400078777774 92ABABAEADADACABAAA9A6A6A3A3A09F9E9D9E9E9D9E9DA0A1A2A2A2C1

:20246000A2A3A2A3A0A0A09FA09FA0A0A09F7B6A6F747575757678777979 7C7D7D7D7C7C2C

:202480007B7A7A79797777766D686665646463626261616 0615F605F60605F5F5D5E5D5C55

:2024A0005D5C5D5D5E5E606162646467676A6A6C6D6F6F727274747678787A7A7C7C7E7F8E

:2024C000808081818383858586878788878887878686848484838280807F7F7D7D7D7B7CA8

:2024E000797A787A77797586A4A7A9A8A9A8A8A7A5A5A3A1A19F9F9F9E9F9D9F9D9E9EA1D1

:20250000A1A0A1A0A1A1A09F9E9C9D9D9C9C9E866D6E686D74777778777A797B7C7D7E7C86
```

```
:202520007D7C7C7B7A7A797978736B69696768676767666666666666666666656665666508

:202540006364636463646466666868686A6B6C6D6D6F707072727475767777879787A7A7C7BAE

:202560007D7D7E7E807F8180828283838484858583848283818281808 07E7F7E7D7D7D7C4B

:202580007D7C7B7B7A7A7A797A787F9CA2A3A5A4A5A4A4A1A29FA09D9E9D9E9C9D9D9D9CD6

:2025A0009C9B9E9D9E9D9E9D9D9C9A99999A979A8D726E6D6B6B6B7478787A797B7C7C7C02

:2025C0007E7E7E7D7E7D7D7D7B7D78706E6D6D6C6C6C6C6C6B6C6A6B6A6B6A6C6A6B6C6BC8

:2025E0006C6C6A6A6A6A696A6A6B6C6C6C6E6E707070727273747476757776787879 7A79D0

:202600007B7C7B7D7C7D7D7E7E7F808081818182828282828081817F807F807E7F7E7DD3

:202620007E7E7C7D7C7D7B7C7C7B7C7A7B7C92A09EA2A1A1A1A0A09F9E9D9B9C9B9B9B99C1

:202640009A9A989999999A999A989A9899959694969 1796F706D6D6C6C6C73797A7B7C7D7D

:202660007C7E7D7F7F8080807F7F7E7E7671727070706F70707070706F706F706F706F70A7

:202680006F707170716F6F6F6E6F6E706F70707171727274737575757676777877797979E9

:2026A0007A797B7A7C7C7B7D7C7D7E7E7D7F7E7F7E80807F8180807F807F807F7E7F7F7E5F

:2026C0007F7E7D7E7D7E7D7E7C7D7D7C7D7C7C7C7B8B9C9B9E9C9E9D9C9D9B9B9A9A9899A3

:2026E00097989798979696959696979596959594939192807273 6F706F6F6F6F6F747A7CE9

:202700007C7E7E7F7E7F7F818181808181 7A757474737372737273737273727373747374AF

:20272000737474737473757474727372727372747374757475757676777777787779787802

:202740007A797A7A7B7A7C7B7C7B7D7C7D7C7E7D7E7D7E7E7F7E7F7F7E7F7E7E7F7E7F7EDD

:202760007F7F7E7F7E7D7E7D7E7D7E7E7D7E7D7D7D7D7E7B859698989A999A9A9999989875

:20278000969796969594959394929393929492939291919086757572737272717272727242

:2027A000747B7E7E7F808080818082828282 7F777876777675767676767576767676767 7CE

:2027C000767776777677777777877787777757676767675777677777877787879797 87A7916

:2027E0007A797B7A7B7A7B7C7B7C7C7B7D7C7D7D7C7D7C7E7E7E7E7E7E7E7F7E7F7E7F7E46

:202800007F7F7E7F7D7F7D7E7E7D7E7D7E7E7D7E7D7E7E7D7E7E7E818F9595979697959640

:2028200095969495939493929391929190908F90908E8F8D8F897A767473727271727 1729F

:20284000717271737 97B7D7A7D7F81817E807D807F7572727372737371706F6F6C6E6B6DB9

:202860006A6B6C6C6C6B6C696D676B6B646866676A68656266646465625F5B65665F636384

:202880006369646A696A67686E6D6F6FBAD8CBCFCECFCACAC9C8C5C7BFC1C1C9CDCACAC797

:2028A000C6C9C6C3C2C2C2BCBFB7B8B4A9A1A29F9C9B9A979695959191938E5E4F57514D4DC8
```

```
:2028C0004E51535153515352545356595857595959595C5C5A5E605E5453535252515353505547
:2028E00056565657555575C5F616265658AA19DA3A3A2A1A29A9A9A999C9A9A999A795E60F7
:202900005A5B5958585958575B5B5B5C5A5C5C5D5E5D5E61626065646766666768686967D6
:202920006A6A6C6E6D6E706F6F6F727171747196B6B5BCBCBFBEBFBFBDBBB9BBB6C3EBEF65
:20294000EEEFEBEBE7E7E1E2DCDDD9D9D4D4CBA79C989592908F8D8D8D8A8C6B49494A53DD
:2029600051525353545556565659595B5A5C5C5E5D605F6160626363655F5659565857595A0C
:202980005A5B5B5D5E5E60606063636A6A6C6B7E9192949696969795918F908E8F8E8D8C8EEE
:2029A000806A696866666666666667676869686A6A6A6C6B6D6C6E6D6F6E706F71717272A6
:2029C0007373737473757475757676777787679767A758BB0B3B7B7B9B9B9B9B7B7B6B5B4E8
:2029E000B4C4CBC9C9C8C8C4C5C2C2C0BEBDBBB9B8B6A69B9B999696939490958158B5854D3
:202A000053555D5F5F616063636464666656767676868696 96A6A6B6A6D6960605F605F6045
:202A200061606263636365646665676768686F6F707175828384858586848 17F807F807E4A
:202A4000807E7F7D71706F6E6F6E6F6F6F706F7070717171737273727474747575747676 7512
:202A6000777677787878777979797 97A787A797B7A7A7A7C7983A6AFB1B2B3B3B3B3B2B102
:202A8000B0AFAEAEB2B7B5B5B4B3B1B1AFAEADABABA9A9A7A7A19A9A98979695958F6C60C0
:202AA000605C5D5B5C65696A6A6B6D6C6E6D6E6E6F6F706F717170727173726866656565D5
:202AC0006665676767686869696A6B6B6B6C6C6D6E71747475767B7D7D7E7E7F7D787978DC
:202AE00079797979797979757374747374747375757475757675767677777778777879780D
:202B000079787A7A797A797A7A7B7A7B7B7B7B7C7C7C7B7C7D7C7D7C7E9BAAAAADADAEAD10
:202B2000AEACACAAABA9A9A9AAA9A8A8A6A6A3A4A2A1A19F9F9F9D9D9B9997979595957A73
:202B400068676464636362636A7071717373747375757575767676767777777770 6B6A6A93
:202B60006A6A6A6B6A6C6B6C6D6D6D6E6D6F6E706F707170747776787 77A797B7A7B7B7A28
:202B8000767776777777777677777775777776777677777877777978797879797A797A7947
:202BA0007A7B7A7B7A7B7A7C7C7C7C7B7C7C7C7D7D7C7D7D7D7D7D7D7E7D7F7D8FA4A5A604
:202BC000A7A7A7A6A7A6A5A5A3A3A3A3A2A2A0A0A09E9E9D9B9C9A9A98999798959592951E
:202BE000846E6C6A6868676867686 76C7575777678777879787 97A797A797A797B766E6F7D
:202C00006D6E6E6D6E6D6F6E6F6F707070717171727173727374737578787A797B7C7B7D5C
:202C20007C7D7A797978797978797879787977777877797879787979797A797A797A797B74
:202C40007B7A7B7A7B7B7B7C7B7C7B7C7B7C7C7D7D7D7D7C7D7D7D7D7D7D7D7D7D7E7D87E5
```

```
:202C60009C9FA0A2A1A2A1A2A0A19F9F9D9E9D9E9C9D9B9A9B9999999797979596939592DE

:202C8000938B75706F6D6D6C6B6C6B6C6D6B6F767A7A7A7B7C7C7C7C7B7D7C7D7C7D7B7360

:202CA00071717071707170717072717273737374747374757475747676757777A7B7B7B7C91

:202CC0007D7E7E7E7E7C7A7B7A797A7A7B7A7B7A7B7979797A7A797A797A797A7B7B7B7A9F

:202CE0007B7A7B7C7C7B7C7B7C7B7C7B7D7C7D7D7C7D7C7D7D7C7E7E7D7E7E7D7E7E7D7E45

:202D00007E7E80959A9B9C9D9D9C9D9B9C9A9B9A989A99999998989795969595949393929F3

:202D200092918F7B737270706E706E6F6E706F706F71787D7C7E7D7E7F7E7F7E7F7E7F7F6A

:202D40007873747274737473737474747574757575767576767677767777777877787A7CC1

:202D60007B7D7C7F7E7E7F7F7D7C7C7B7C7C7B7C7B7C7C7B7B7A7B7A7B7A7B7A7B7A7B7BDA

:202D80007B7C7B7C7B7C7B7C7B7D7C7D7C7C7D7C7D7C7D7C7D7D7E7D7E7D7E7D7E7E7D7EA1

:202DA0007D7E7E7D7F7E8D969697989797989697969695959496959495939493929192920 6

:202DC00091918F908478767474717371727173717372737373797D7E7E7E7F7E807F7E858F

:202DE0008588827E807E80807E828184838586838688878685 8C7A7373727168736C6D6E21

:202E00006A6768696C6E6A6F5D645B6B6C5B5B685A5F625B6C546860625B5D686671646F22

:202E20006A616A6D6273716279706E6D6E70737587B1A6ADADB5B0ADB1A9B0ADAC9DA09F35

:202E4000A5A097A09F979FA09BA2B3AAB1A87D7D7C777A7A7C7B7676777274777276706E30

:202E60007974787572766A615A5F625F615B62645E6C66626664646C6D717073736A717816

:202E80007584B8D2D3CAC8CFCDCCC6C3C1C6C1C6C7BEC6C4BEBDBEB9BABCBEB5B9B8B6B981

:202EA000B1AFB8A985654A4540403D3B3B3B373B3C383C3A3A3E4043474346464B4B504DCA

:202EC000505451554E54565857575B5A575C5961635F60678C8F9095959595969A96939314

:202EE000969792949191918F8E938E8E91A0A1A1A39FA19F9D9A9D9CA09F9999A1856B6D2C

:202F00006B6A65686762635D575754545954585B5A5D5A5C5E5F605E636165636C6D6C6DAA

:202F2000716F727572ABD0CDCCCACBCACDC8C5C9C5C4BFC0BBBABCBAB8B6B8B3B2B3B0AE33

:202F4000ADACA7A8A6A8A68C7D7C7777767576777473717478 5C454847434543474945472F

:202F6000474B4C4B4D4E52555451565654565956 5A585C5B5B5E5C808B8B8F8D90929190C9

:202F80009291928F90918F908F8E8E908F8E8D8F9A9C9D9F9D9C9A9C99999A9D9A9799978E

:202FA000979796979595959389808380816053575151525350535455585556595A5A5C606E

:202FC0006265676567686B6981A8A7A6A6A6A4A7A5A5A2A3A1A09FA09E9E9F9C9B9B9C98BE

:202FE000999697979395919092918A817F807E7D7E7D7E7C7E7B7A7C797A7A7B7B7B787C4D
```

```
:203000007A7C7C7D775A5154555352545354545358585658606566B6E716E849D969B9B9D47

:203020009E9D9F9E9F9B9B9A98999A9A949896988D6F71767D7D7D7A7B7E7B7E7D7F7E7D2D

:203040007F7C7C7C7D7D7D7D7E7A6D6B6B6D6B6C6C6A6B6C6B6C6C6E6E6D6E6F6D6F6E6F3B

:2030600070737379787B7A7C7D7B7C80A1A8A1A1A3A2A2A49F9F9F9F9FA09D9D9C9D979831

:20308000979797959594939583 6E706E6D6A6767656765666867686465676769646969 6AA3

:2030A000696B6B6E6C6D6E6B6E6F6F7170717072737172727376727373757275728094 97AF

:2030C00099989B989B99999896989895959396949293919392929 6A09FA2A0A29FA3A0A0D2

:2030E0008D7C7E7B7C777976797976696567656467666564666566676968686969 6C6B6B0B

:203100006B6D6B6D6C7074747778767778787 98C9A929694959395949392909390928F9139

:203120008E8E8F8E8D8C8D8D8B8D8B8C8C8C8A8B8A8988878A87868873686968676566 6444

:203140006665686769676868 6A696C686B6C6B6B6F6F70726E70717172717273737 27375CC

:20316000748C8F8F909091929390908F8F8F8E8F8F8F8E8B8D8D8D8E8C8F9C9C9F9E9F9D39

:203180009D9D9C9B999B9A99979895978B83847F6A656665656467646465666466666667 91

:2031A0006767686A6A6B696C6D70747273747775757481918A8C8C8C8B8C8C8B8D8C8B8CA8

:2031C0008B898B8A8A89898889878687868787858787878686838483848483828482838 12A

:2031E00082818182818 0817F6D6A6C6A6C696B696A6A6B6B6D6E6E6D6E6F6F6F6F70707197

:2032000071717473808A86898789898A8B8989888A89898888888889868886868693999AD9

:20322000999A9999999798979596969595968B817F7F7F7B7E7D7C7D7C7D7B7D7B7D7A7E778E

:203240006A6A6A696A6969696A6A6B6B7071717372737373737786838484878C8D8D8D8EA5

:203260008D8E8B8D8B8D8C8A8C8A8D81787B787977787778787877787776777776787 67871

:203280007678777876787878787879 7A797A7A797B7A7A7A7B7B797B7A7B7A7B7A7C7B7A02

:2032A0007C7B7D7C7C7C7B838B8A8C8B8B8C8B8C8A8B8A8B8A8A898989888A87898787883D

:2032C000909B99928C8F8B8D8B8D8B8B898B8577767473757474747474757574757475 76DF

:2032E0007576757776767776777678767878777A7D7D7F7D7F7D7F7E80888787888888873E

:2033000088878887878687868687868586848584858585858484838482838377767676 7551

:20332000767774767576767677757777787778777977797879787979797A797A7A797B96

:203340007B7A7B7B7A7B7B7B7C7C7D85848585868586858685858585858385848485 83843E

:2033600083838288969797989798979896979597 8D827E77767674747473747374747473B5

:203380007474747474757476 7A7D7E7F7F8180818586868A8B8D8E8F8E8D8F8E8F939497C0
```

```
:2033A000939797979A98999B9D9D9DA2A2A09FA09EA3A8A7ABB694645B59595B4F4B4D47D6
:2033C00049464A484C4C4A4C514F51524A4D50575056514F575854585B58655E5E61626283
:2033E000606B6D74786D78777478737878787D7F8280736C6F6E706A686B6E706D6F6F6F87
:203400006F6970757171747B838484878084838185 7E867B7372707372727473757675739A
:2034200072707371736F7370746D7DC8D5CED1CBCCC6D3F1F5F3F4F1F3EFEFEEEFE8E9E614
:20344000E5E2E1D8AFA5A09A9B96949290908F8C8E8B8D8B8A888A805343443F3E403F3F6A
:20346000414141444346484949484A4C4E4F50535454545457575B5C5B5C5C5D5E5E605F610C
:203480006263676185A5A1A7A4A6A4A5A5A4A2A0A1A0A19CA0896B6B67656565636463656D
:2034A00065676866696967696B6B727574777879787978 7A7A7B777272717173727373 73E4
:2034C0007473747574747574757677777 8789EB4B4B9B9BBB8BCD5DDDCDFDCDDDADBD6D654
:2034E000D4D1CFCDCAC8C5AEA09E9B999796949492929091918F908E8F69545350504D4F61
:203500004D505050515352545456575759595A5A5C5C5E5D5E6060606262646465656766 52
:2035200067696 86B6A6C6B738586898A898B8A8B898B898B898A89898472716F6F6E6E6D12
:203540006D6D6E6D6E6F6E707071717172777C7D7E7E807F807F807F807A777676757675A0
:203560007675777776777778777978 7A797B787C7890AFAFB3B3B3B4B3BDC3C3C3C3C2C19A
:20358000C0BEBDBBBAB7B6B5B3B3A99F9D9C9A999797969595959294 90947D5E5E595A582E
:2035A00059575958595A5A5C5B5D5D5F5E605F6161636264656566676768686 9686A6A6BE7
:2035C0006C6B6D6D6D6E6E6D6F6E71787C7B7D7D7E7E7E7D7D7E7D7E7E7D7F7D7773747318
:2035E0007473747375757475767576757776777 67A7F8181828183828383828 37C7A7A7983
:203600007879787 87A787A797A7A7B7A7B7A7C7B7B7B7C7A87A5AAABADAEADAEAFB3B3B225
:20362000B2B0B1B0AEADACABA9A8A6A6A5A19E9B9B999997989596959592958 96B64636264
:2036400061605F6060616262636364636564666667686769696 96A6A6B6B6C6C6D6E6D6E9A
:203660006F6F6E706F717070727172727373757877787777979797979 7A7A7A7A7A7A7B79A1
:203680007877787778777879787978787979 7A797A7A7A7C8084838483858485858583 7F94
:2036A0007B7C7B7C7A7C7A7C7A7C7B7C7C7B7C7C7C7C7D7C7D7D7D809CA5A6A7A8A8A8A83E
:2036C000A9A8A8A7A7A6A4A5A3A3A2A09F9F9E9C9C9A9999979795969594949391776B69C7
:2036E0006766666666 65676567676768686969 6A6A6B6C6C6B6D6C6E6D6E6E6F6E7070707179
:203700007072717273727372747374747574757675767877787879797 97A7A7A797A797BED
:203720007A7B7A787A7A7A797A797A797B7A7B7B7A7B7A7B7B7C7C808484848584858584EA
```

```
:2037400085837F7D7C7C7B7C7B7C7B7C7B7C7B7C7D7C7D7C7D7D7C7E7C7E7C92A09FA3A129

:20376000A3A1A3A3A2A2A1A1A0A09F9F9D9D9D9B9B999A9997959594949391939092806F31

:203780006E6C6B6A6A6A6A696B6B6A6C6B6D6D6C6E6E6D6F6E70706F717170727271737275

:2037A000737274737473757475767675767677767778777878797A797A7B7A7B7B7A7B7B2B

:2037C0007C7B7C7C7B7C7A7B7A7B7A7B7A7A7B7A7C7B7C7C7C7B7C7C7C7C80838484858451

:2037E00085848585847E7D7C7D7D7C7D7C7D7C7D7C7D7C7D7C7D7C7D7D7D7D7D7E7D889BE0

:203800009A9D9C9E9D9E9D9F9D9E9C9D9B9B9A9A989898969795969393929191908F9087B2

:203820007372706F6E6E6E6E6E6E6E6F6E6F6F706F71717171727272737372747473747473

:2038400075747576757675767677767776787778787779797879787A7A7C7B7C7B7C7B7D67

:203860007D7D7C7D7D7C7D7C7D7C7B7C7B7C7C7C7C7B7C7B7C7B7C7C7D7C7D7C7D7E8383B4

:2038800085858585848585847F7E7D7D7C7D7D7C7D7C7D7C7D7C7D7D7C7E7D7E7D7E7E7F47

:2038A0007D8294979799989998999A99999898989697959695949493939390908F8E8E8E91

:2038C0008A7A73747172707170717071717172717273737274737473757574757576767670

:2038E00076777777777778787877787879787979797A797A797A797A7A7A7B7B7D7C7C7DA5

:203900007C7D7D7E7D7E7E7E7E7E7D7E7C7C7C7B7C7B7D7C7D7C7D7C7C7D7C7D7C7D7C7E0F

:203920007D828483858485858485837F7D7E7D7D7D7C7D7D7C7D7C7D7C7D7C7D7C7D7D7CA5

:203940007D7C7D7C7F8C939295939593959695959495959394949393929291929091 8E8E7D

:203960008D8E807978777776767675767576757676767676767577767675777677757675 4B

:203980007675767575767576757676757474757374737576757776777877797 87A7A7B7D5E

:2039A0007C7F7D80808081818283838484848586848585868686888888888887878585848A

:2039C0008383828285868784858382828 07E7A7978777475747373726F706F6F6E6C6C6BB8

:2039E0006B6B6D6D6E6F70727C86868A8B8D8C8F90939495969797999A9B9C9D9E9F9FA17F

:203A0000A1A1A19B91919090908E8E8B8B898986868383807F7D7C7A787776747271 6F6EF5

:203A20006D6B6B6A686866666465636363616261626262646666696A6B6E7070737576786F

:203A4000797C7F808383868789898C8C8E8E909092929392939394949594949292908F8ED4

:203A60008B8B88888585838385838381807F7C7C79747370706F6E6D6C6C6B6A6A69686734

:203A8000686768686A6C6D6F6F72737A86878B8B8E8E91929596969897999A9A9B9B9C9C7A

:203AA0009D9E9E9E9B8E8D8C8C8C8B8A89878684848282807F7D7D7B7A7878767674747359

:203AC0007270706F6F6E6D6D6C6C6B6A6A696968686768686A6C6D6E6F7171737476767930
```

```
:203AE000797B7C7E7F828384858787888888A8A8C8C8D8E8D8F8E8F8E8F8E908F8D8D8B8BB3

:203B000088888686858383818 17F7E828080807F7E7C7C78767474727272707070 6E6F6D42

:203B20006E6D6E6D6F6F717272747675787 77C888D8F909192939496979798979 9989A9AFC

:203B40009A999B9A9B998F8A878786868584838281817F7F7D7D7C7B7B79797877777576D9

:203B600074747373727272707170 6F706E6F6D6E6D6E6C6E6D6F707072727475767678781F

:203B80007A7A7C7D7D7E7E808284838585868787888889 8A898B8B8B8C8B8A8B898988887D

:203BA00086868484838381818 0807E7E7D7D7E807F7E7E7D7D7C7A76777676747574737 36E

:203BC0007273717272727474757577777979 7A7A7C7C899091939294939595979698979887

:203BE00097989898999799 8F87868684828281 7F7F7E7E7C7D7C7B7B7B797A787978777748

:203C000077767775767574747374727372737173727272717172717373757575777878 7824

:203C20007A797A7B7B7D7C7D7D7E808081808281828381838284848385848684858586856E

:203C4000848385838381828180817E7E7E7D7E7A7C7D7E7F7D7D7C7B7C79787778777776B7

:203C60007674747575747475757984898B8D8E9091919092909AA4A5A6A7A7A9ABA9A7A976

:203C8000A8A9A9AAA5A8A6A8A29699958470726E6E6C6C6B6A69696A696A696A6765666A12

:203CA000686765676565676767686969656 86B6667676667686766686768686565676B6B6C15

:203CC0006C6D6D6D6F6E6E6F6D7170707171706F73717372757575757677 7A797978797B96

:203CE0007E7D7C7A7C7B7B7B7D7B7C7B7B787C7A7B7B7C7B7B7D7E7C7D7F7D7F7D7B7C7B48

:203D00007B7A7A797B787C797A7B7A7C7AA5C4BEC1BCBEBBBAB8B7B4B9C5C3C3C1C2C1C06C

:203D2000BEBDBBBBB9B7B7B4B5B1A49D9E9A9B886E6A686666656463646463646465 65665A

:203D4000666567666868676868696969 6A69696B6A6C6C6B6C6C6C6D6C6E6D6E6E6F6F7013

:203D60007171727174747475757675777777797879797A7B7B7B7B7C7B7D7C7D7E7E7E7E3A

:203D80007F7F7F7F80807F807F7F7E7F7E7F7E7F7D7E7E7D7E7E7D7E7F807F807F807F7E4C

:203DA0007C7D7D7D7D7C7D7C7C7C7B7C7D7B7D7C85959597979798989797989699A5A9A8A99D

:203DC000A8A9A7A7A7A5A5A5A3A3A1978D8D8A8B888985767371707 16F706F707070706FB3

:203DE00071717071707172717271727273727372737274737473747374747575757475756C

:203E000075767677767877787879797 97A797A7A7B7A7C7C7C7C7C7D7D7C7D7D7E7D7E7E5B

:203E20007E7F7F7E7F7E7F807F7F7E7F7F7F7F7F7F7F7E7F7E7F7E7F7E7E7D807F80807FA9

:203E4000807F7F7D7E7D7E7E7D7E7D7E7C7D7C7D7C7E7D8086898889898A898A898A89951A

:203E60009C9C9C9D9D9C9D9B9B9B999B9286848382828081808 07B777876777677677762B
```

```
:203E8000777677777677767877787877787778777878797879787978797978797787A797A25
:203EA0007A7A797A797B7B7A7B7A7C7C7C7C7C7C7B7D7C7D7C7D7C7D7D7E7D7E7E7D7E7D86
:203EC0007E7D7E7D7F7E7E7E7F7D7E7D7E7D8082838483848486868686878688888B8B9C
:203EE0008A8C8B8D8E8D8E8F8E8E928F909294969493969797 9E9CA1A1A7A6AAAB8A696671
:203F00005D5B64696A686A6869656667695D5B544A4E51555255565A5555615E5D5B5B5BFC
:203F2000635A5B5F5F605A646162636262676669686867686F6D6F6C70706F6D6C6F6E72A0
:203F40007272717474757273707574757774777875787977797 87A7B7C7A7B7B7C7A7D7B85
:203F60007C7C7D7E7D7F7E7F7D7F7D807D817E837AA8E6DEE0DBDBD6D5D2D0CECAC9C8C5A6
:203F8000C4C2C1C0BEBDBABAB7B7B5B3B3B2B0AEAEACABA9A9A8A7A7A5A4A3A4A1A48B6283
:203FA0005D5A575655585E5F606062636264646665676 15D5E5E5F5E606161616261626310
:203FC00062646564666667666868696 96A6A6B6C6B6D6C6E6E6E6F6F707071717172727277
:203FE00073727473747574757576767677767777787877797879787A797A797B7B7B7A7BDC
:204000007A7C7B7C7B7C7C7C7C7D7D7D7E7D7E7D7F7E7F7D89B3BABDBFBFC0BEBFBEBCBC3E
:20402000BAB9B8B6B6B3B3B1B1B0AFACACAAAAA8A8A7A5A4A3A2A29FA09E9F9D9C9B9D946E
:204040006B5F5E5A5B585959596063646565676668676 96A696462626163626363646 46516
:204060006465646665676768686969 6A6B6A6C6C6C6D6D6E6D6F6E70707070717 2717272CB
:2040800073727473747475757476757675777777777 87778777978797 87A797A797A7B7B46
:2040A0007A7B7A7B7B7B7C7B7C7D7B7D7C7D7C7E7C7E7C7E7E7D7E80A4B4B3B7B7B7B7B6BB
:2040C000B6B4B4B3B2B0B0AEAEACACAAAAA8A8A7A5A5A3A3A2A0A09E9E9D9D9B9B999A9914
:2040E0009877646460605E5F5F5E5F5F65696A6A6C6C6D6D6E6D6F6E6867666767686768AB
:2041000068696A6869696A696B6A6C6B6D6C6D6E6E6F6F6E706F717070727272737 27374EB
:20412000747474757475767576757776777678777778777978797 87A797A797A7A7A7B7A95
:204140007B7A7C7C7C7B7C7B7C7C7C7D7C7D7C7D7D7D7C7E7D7E7C7E7D7E7D95ADABB0AFF4
:20416000B1AFB0AEAFADADABABA9A9A7A7A6A5A4A4A2A2A19F9F9E9E9D9B9B9A9A999798C1
:20418000959883696866656463646365646564696E6F6F7170727273737 2726C6B6A6B6B58
:2041A0006B6B6B6C6C6D6C6D6C6D6E6D6F6E70706F7070717172727373747676767 77678F9
:2041C0007879797979797A797B7A797677777776777678777877797879787 97A787A7879D5
:2041E0007A797A797A7B797B7A7B7B7B7B7B7A7C7B7B7D7B7C7C7B7C7B7C7D7C7E7B89A129
:20420000A4A6A7A9A8A8A7A7A5A5A6B0B3B2B3B1B2B1B1B0AEADADAAABAAAAA89992918F95
```

```
:20422000908B8E8268615E605D5D5E5D5E5E5E5F605F6367686B6C6B6D6C6E6E6F6B68684C
:2042400067656669666869 6A676868696A6A6B6B6D6C6A6B6B6B6C6E6C6E6C84A8A7AB4B
:20426000ADABABABADACACA9A9A8A9A5AB9A716C676663636360606063606462656565465EF
:2042800065686868696969 6B6D6C6F6C6D6F6F706F7070727273727473747475747775 7739
:2042A000757C939B9B9C9B9F9D9E9C9D9D9CBCD3CFD3D1D2CED0CCCCC9C8C7C3C3BFC1A3B6
:2042C0008A8B84847F685C5D5A5B5959595B5A5C5B5C5E5E5F5F66686A6B6C6D6B6D6D6FEF
:2042E0006C6969686969696A696B6B6C6B6D6D6D6E6E6F6F707071727073727373747379FE
:204300009499999C9B9D9B9C9B9B999A97999698927C73736F706E6F6F6E6F6E706F7070DC
:20432000717272727373737473757574757576757776777777787778787978797A797A7AC1
:204340007B7A7B7A7C8E9A999C9B9C9B9C9B9B9B98A4B2B1B2B1B0B0AEADADACABAAA9A89F
:20436000A6A79C8F8C8A7A6B6A686767666667676668676968 6A696B6A6F727373757475BD
:2043800075767674706F706F7071707170727172737372747473757475757676757676 77B9
:2043A0007677778388898A898B898A898A898A89888987887E7878777677767776777677EC
:2043C0007678777877787979797878797A797A797A797B7A7B7A7B7A7C7C7C7B7C7B7C7B7DA3
:2043E0007D7C7D7C7C7D7D7D88979699999998999997989 6999FA09FA09E9F9E9E9C9D9B65
:204400009A9A9A9799949084757270706E6E6D6E6E6D6F6F70706F7070717171737979795F
:204420007A7A7B7B7A7C787575747574757475757675767776777677777877777878797 89A
:2044400079797A7A7A797D8081808182818281818081808181 80817F7A7C7A7B7B7A7B7AA5
:204460007B7A7B7A7B7B7C7B7C7B7D7C7E7D7D7E7D7E7E7F7F807E7F7F7D7E7D7F7E808196
:204480007C82838384867E817E878A899795998285857C89817F7A878A7B85788674757EB1
:2044A000717067716DA4B6A2A4A29191938B898E9783808A9285797D89767375707C7F6A50
:2044C000424E4D44424750434C4C534B4D5659695B66655B635662626167675C61676770DC
:2044E00071756E6B6B68616866716B626661635C64606363756A75969A9D92999098979012
:204500009D98B3ECEDEFE7E2E7E3DEDADADFD7D4D8CFD6CAC6BFCCB2959B899190918B807C
:204520008C8A8887848785817D8585828480848A64494D4B494942424B484E4D4747444986
:2045400045464E4D554B535273887B7C787D787D7A7A7F7B7D7D7E7B7D827F7E7D7E7C7B45
:20456000807C7C7F808381867F7F827DA6C2BE9A8B938B888688838485827D7F7F7D7E7C3E
:204580007E7C807D7C7C7B7D82807B7E7A806B4F4D4D4D4B4B4F4C4D4D587A817F8686886D
:2045A000878B8A8A858BB5C8C5CCCACBCDCAC7C7C7C9C2C1C2C1BDBDBCB9B3B7B3B5B2B044
```

```
:2045C000B2B0ADA9AAA9A480777877727070727071593D433F4242444142414747454B4897

:2045E000494C4C4F524F5254545755677B757A77787B7E7F7C7B808180807F7C7F817D821A

:20460000807F7C7D7F7E82808381838380867F8B999A9B9B9B999C99979A939594938F6B3D

:20462000656461626161635F5F63636363626463676351505052544F545356555778838 37F

:204640008A898B8C8B8B8A8B8BA7BFC2C2C4C4C4C2C2C1BEC0BBBCB9BBB9B5B1B1B4B1B0B1

:20466000AFADAAACAAA9A7A9A5A5A1A3A0A1A2A09CA18D6C646B64454644454445444646FE

:204680004447484B494A4D4D4E4F53524F475170706F777377757879787879787879777ACD

:2046A000797B787A7B797A7B7A7C7E87826A6F6B6A6B6E737373737574747678747577753B

:2046C0007575757475767876777478777979787777797679787 16F6F716E72727371716F46

:2046E000869A989A9A9C9B9D9E9A9B98A9C3C9C9C9C8CAC7C6C5C2C3C0C0A5989B94919052

:204700008D8C8D8C8D8A8C8A888B8A8A898989878987878065585953595E62636065636667

:20472000666768676768 6A68686C6C6C6B6D626161617F83838686898787888986868887DF

:204740008583878586858684878586878785878785856863646165626465636664666668 72

:20476000676667676 96A6A6C6C6B6E6D706E6E6F6E6C6F6F717172716F6F707170727273 85

:204780007274707E949297949699989595 98939CB6BEBCC0BEBFBFBEBABBB9B8B8B5B5B298

:2047A000B3B1AEAEACACACA7A9A28A87878483807F7F807F806655565451 53565D636262C4

:2047C0006464666668676A686868686B6A686D645F645F605E74817E837F838385848484F2

:2047E00085858483838481838182828381828181848484838485828380818481858281 8263

:204800008477686765656665646365676668686769676969696C6B6C6E6D6D6D6E6B6F6F4F

:2048200071706E6F737075888A8B8D8E8F8F8F8C8F8C8FA5AFAFB0B2B2B0AFB1AEADACABEE

:20484000ABA9A8A5A7A5A4A5A3A3A0A09F9E9D9C9A9B9A98989A8C74756756595857545DA2

:20486000636364656366656868696A6A6B6B6D6B6A626063636163626D7A7A7B7D7E7F7FD3

:204880007E7E7E7E7E807E7F7F7E80808080807E807E81818082818182807F807F7F7F2A

:2048A000807F7D807E807E7F7E7F7F7E807F817F80746A6B6A6A696C6A6C6B6B6D6D6D6D36

:2048C0006D6F6E6F6D6F6F71717080858486868785888788888594A4A4A7A6A7A6A6A4A4D9

:2048E000A3A3A2A2A19E9F9D9E9C9C9A9B999A979B978E8D8E8C8C85716D6C686B6A6A690C

:2049000069696C7575777778787878797879787A7A77716E6F6E6E6F6E6F6D7280807F81FF

:2049200081828282838383828282818280828180818081807F7273727473737472747172EC

:204940007474737574757576757676767876777778787778797978787A787A797A7A7A7A72
```

```
:204960007B7A79797A7A7A7A7A797A7C798187878888888898888898889868E9FA1A2A2A3A257

:20498000A1A29FA19E9F9D9E9C9D9A9B968C8D8B8A898B888888888787766E6E6B6C6A6CA9

:2049A0006A6C6B6B6C6F7578777A787A797B7B7A7C7B7C737171717071717172717273797A

:2049C0007D7C7E7D7E7E7D7E7E7E7F7F7E7F7E7F7E7F7E7E7D7F7E7F7F7E7F7F807F807F0D

:2049E000807F7E807D7678777677767777777877787877797779797879797A797A7A7A7994

:204A00007B7A7B7A7B797A797A797B797B797A7A7C828182838282838283828385939898 99

:204A2000999899969896959597949491928F919091908F8F8F8E8DA0A6A59D928F8A8B8DF8

:204A400091908E8D8A8F8D8C91989D9FA3A5A8A7A8A2ADAAAEB1A8A7A3A8AAA9A6A0A4A174

:204A6000A4A1A2A2A48E534343424341403E403D464547434643484B454C494F4C514F504B

:204A80005653636064686A6C6C6D6E6B716F6D726C6F736B5F5C5B5E5A5E59595D595D5D90

:204AA0005F5E5C5D5F60616063626669686B686C6A6A696B716F6D707070706F70727071DE

:204AC0007B868483828885838687838784878486838683 93B7BABEBDBFBDE4FBF1F6F1F55D

:204AE000F1F2EEF1E5BAB0AEA5A4A29F9D9E9D9C9B9A999A98979796949 08E8E8D8D8D8CC2

:204B00008B8A898B888985584745414241414042424445464 6494A4C4D4D4F4E51505454B5

:204B20005657575A5977939296959895989796979694949492967D61605D5C5C5B5C5B5D32

:204B40005E5E5F5F6061606362646364666666768696A6B6A6C6C6D6D6E6D6F706F71717159

:204B600072727378828284848486858686868687868686868686 88A3A9ABABADA8B2D5D929

:204B8000DADAD9D9D7D7D4D3B9AEACA9A6A5A2A3A5A4A5A2A3A1A2A09F9F9D9C979694942C

:204BA000929392928F918D916F53524E4D4C4D4C4D4C4D4F4F5152535355555758595 95BA7

:204BC0005B5C5E5D5F5F6160697A7B7E7E807F8181808180817F8180817C6B6A6968676895

:204BE00067686969696A6A6B6B6C6C6D6D6E6E6F706F7171717271737273737475757476DC

:204C00007576767777767B848687878888898889898989898888888888888919898989289CA

:204C200091B5BEBFC0C1C1C0BFBEBFB4ACAAA9A6A6A3A4A6A7A5A5A3A2A3A1A1A09E9D98F9

:204C400097969594949194909482605B595856575556575658585A5A5A5C5C5D5E5E605FE0

:204C600061616363636564666667666A71747476757777767777787877787779787 16F6F51

:204C80006F6E6F6F706F7170717271727273747374747575757676757776777777787778A6

:204CA000787978797A7A797A797C8588898A8A8A8A8A8A8A8A8A898A89898989888B908B61

:204CC000828280819DAFADB2B1B1B1B0B0AFADA8A6A6A3A3A2A0A4A4A4A3A3A1A1A19F9F86

:204CE0009D9B969795949493929 28D6C63615F5F5D5E5D5F5E5E5F5F616261636264646559
```

```
:204D000066676668676968 6A696B6B6C6B6C6D6F7171717272737374747475747574767 5C2
```

```
:2050A0007578787878787879787A7A7B7C7B7E7E7F7D8079706D6E6E6E6C6E6F716F6F727238
:2050C00071727273737474757476767777778777A787DAABCB6BBB6B9B1CAEFF1F1F2F2F026
:2050E000EFECEBE9E5E5E2DEDDDAD9D6D4D1CECCCBC9C6C6AD95948E8E8A8A888887878608
:205100008685868484 5C4545414140413F424243444646484 84B4B4D4F4F51525254545633
:20512000575D5F6061626264656567666764626464646564666667686869 6B7479797B7A88
:205140007C7D7C7D7D7E7E7D7E7D7F7F7E80808180818080757371717071717072717272 1B
:205160007273737475747574767577767776787778787979798593919493959 39DC4CDCE8E
:20518000D0CFCFCDCDCBCBC9C7C5C3C2C1BEBEBBBBB8B8B4B5B1B2A99B9897959393919158
:2051A0008F908E8F8D907657555251505050515052525455555657585 85A5A5C5C5C5E5FBC
:2051C0005F6060626569696B6A6C6C6D6D6E6E6E6B6A6B6A6B6B6C6B6D6C6E6D6F6E777D6A
:2051E0007E7F807F80808180818081818282828282838382837A757474737373727373741E
:205200007374747575747675767776777678777879787979787A797A7C8484868685868848
:20522000A7B8B7BBBABBB9BAB8B8B7B5B4B4B1B1B0AFADACAAAAA9A7A7A5A49C9A999797C6
:2052400096959492949194876760 5F5C5C5A5B5B5C5C5C5D5D5E605F6160626264646465CE
:2052600065676668676969 6B706F717271737372747373707070707070717171717272 7350
:205280007278808181828283828483848384838482848383848 57E7876767576757475744C
:2052A00075767576767677767677777877787978797 87A797A797A797B7A7A7C7D7F7D7FD6
:2052C0007E807E93AAA9ADADAEACADABACAAAAA8A8A6A6A5A3A4A1A2A0A09E9F9C9D9999AD
:2052E000979897969695939490756768656463636363646364646666666767686869 6A693C
:205300006B6B6B6C6C6C6E6D6E6E70747575767577767777777774727372747374737 5744D
:205320007575757578808383848584858485848584858485858485837A777876777677759C
:205340007776777677767878777877797978797 97A7A7A7A7A7B7B7A7B7A7B7A7C7A7C7C2A
:205360007E7D7F7E7F7D89A1A2A5A5A7A7A6A6A5A4A4A2A2A1A19F9F9E9E9C9C9B9B999AFB
:2053800098979595949393929 2937E6D6C6A686868676868696869696A6B6A6B6B6C6C6D16
:2053A0006D6E6E6E6F707070717071717272767778777979787A797A797576757676 75766D
:2053C000757776777777777787F8484858585868586858685868585 86847D787977787778FA
:2053E0007878777877787879787879787A797A797A7A7A7B7A7A7B7A7B7A7C7B7C7B7C7C7A
:205400007C7D7C7F7D7F7E7F7D83989F9FA0A1A1A1A0A0A09F9F9E9C9C9B9B9A9A9998981F
:2054200096979694959293909 28F908671706D6D6B6C6C6B6C6B6D6C6D6C6E6E6E6F6E6F0C
```

```
:20544000706F717071717271737372747473747475777797A797A7A7B7A7B7B7A777877789D
:205460007877787778777979787978 7F8584868685868687868685868586817A7A78797858
:20548000797879787979787A7A797A797A7A797B7A7B7B7A7B7A7C7B7C7B7C7C7B7C7C7DC3
:2054A0007C7D7D7D7C7E7F7F7F7E7E807F919A9B9B9C9C9C9B9C9C9B999A99999898969623
:2054C0009595959393949392928F908F8E8B79727270706F706E706F706F7171707172717208
:2054E000717373727373747474757475767675767577767778 7B7A7C7B7C7B7D7B7D7B79CF
:2055000079787978797878A797A7A797A7A7A7D8585868686878687878686 86847C7B7A7AB5
:20552000797A797A7A7A797A7A7B7A7B7B7A7B7A7B7C7B7C7C7B7C7B7C7B7C7C7D7C7D7D0C
:205540007C7D7C7D7C7D7D7E7E7E807E807F817E8A97979798989897979796969595949413
:205560009493939192919091 8F8F8F8D8D8C7F767574737272727173727372737473737426
:20558000737474747574777878797A797A7979787B797A797B7A7A79797879797A797A797B01
:2055A0007B797778777978777879797979797B7A7A7A828786868789878889878 78A848111
:2055C0007F7C7E7B7B7C777A797878797B797A777879777A7679767C878A8D8B8C8C898D0E
:2055E0008A88898B8B8989897E7271726E6D6D716E71707582858586898988878C8C8C8BAD
:205600008D8D8E8D90918D8F8C9192919191948B827D7C7C7976797974737571727 0706F31
:205620006F6E6C6A6B6C6866658EA7A1A3A2A09F9E9B9B9A98959695919773 5B5C57595833
:205640005B5B5E5F61616265676B6D6F7073737577797A7E7F868B8D908F91909293939444
:20566000948E8A8686848281807E7D7A7B787874757372706F6E6D6B6C68729198979A9761
:2056800097959693929091 8D8E8D8F896B636362636366646868 6D797F81858588898C8C36
:2056A0008D8F8F91919392959496959798989899958985848181 7E7E7B7A7A787775 7573CC
:2056C0007371716F6F6E6C6C6B6A6B677788888989888887858583838382838 3867B6A6B89
:2056E000696B6A6C6D6E70707072737576787A7A7B7D7E8080828384888E8F919292939419
:205700009394929 28D87858482817F7F7C7C7B797977777674757273727170706F6E7E85E6
:2057200084858484838380818080808181828277 6E716F717172737476 76828A8C8D8F919A
:205740009192929494959596969796989897999998988A858482817F7E7C7C7A7A797777DA
:205760007576757473737272 71706E6F6D6F6C717A7B7B7C7B7A7A797979797B7A7C7B7E6B
:205780007B74727474757577777977797 97B7B7D7D7F7F7F8181838384858488 8D8F8F9041
:2057A000919191908F8F8D888482828 07F7E7E7C7C7A7B797977787776767675747473 7306
:2057C00072767C7A7B7A7B7A7979797A7B7B7C7C7D7E7B7678777978 7A7A7B7C7C858F9156
```

```
:2057E000939394949695969697969796979798989797978E838380807D7D7B7C7B7A7A781B
:205800007877777676757574757374727372727172707274767576757674757577767878F3
:2058200079797B7A7879797A7B7A7C7C7C7D7C7E7E7F808080818182838284838584868A97
:205840008E8E8E8F8E8D8C8C8A8B85828180807E7E7C7C7C7B7B7B797A7979787877787624
:2058600077767776767977787877787777787879797A7B7A7C7B7B7A7C7B7D7C7E7D7F7EF6
:20588000848F93939595969696979697979697969696969691848281807E7E7C7C7B7B7B70
:2058A000797A7978797878787876777677757676757575747574767675767576767576751C
:2058C00077777878797A797A797A7B7B7C7C7C7E7D7E7D7F7F7F807F818180828182828329
:2058E0008283878B8B8A8B8B8B898A88898381817F7F7E7E7D7D7C7D7C7B7C7A7B7A7A7A6F
:20590000797A7A7979787879797A797A797A787A797A7A7B7C7B7C7C7D7C7C7D7D7D7E7D36
:205920007E7E7F818D939395959595959594959595959593959185808007F7E7C7D7B7C7B17
:205940007B7B7B797A797A78797879787979777877787878787877777777797879787979 36
:20596000787979797A797B7A7B7B7C7A7B7B7C7B7D7C7D7C7E7D7E7D7F7F7F807F807F80A0
:205980008081808181838887888788878888878782808007F7F7E7E7E7D7D7C7D7D7C7C7BD9
:2059A0007C7B7C7B7C7A7B7A7B7A7B7B7C7C7B7C7B7C7B7C7C7C7D7C7E7E7E7D7E7C7D7E69
:2059C0007D7E7E7F7D7F7E8A929394939594959495939493949293877F7E7E7D7C7B7C7BB8
:2059E0007B7A7B7A7A797A797A797A7A797A797A797A797979787978797879787A7B7B7A77
:205A00007B7B7A7B7A7B7C7B7C7B7D7C7D7D7B7C7B7C7B7D7D7D7C7D7D7E7D7E7D7F7E7FFD
:205A20007E7F7E80807F807F818486868685868586858580807E7F7E7E7D7E7D7E7C7D7D51
:205A40007C7D7D7C7D7D7C7D7D7C7D7C7B7C7C7E7E7D7E7D7E7E7D7E7D7E7F7E7F7E7F7F9E
:205A60007D7E7E7D7E7D7F7D7F7D86919394939594949493949492948B807F7D7C7C7B7B45
:205A80007A7B7B7A7A7B7A7B7A7B7A7B7A7A7A7A7B7B7A7B7A7B7A7B7A7B7B7B7A7B7A7BB6
:0E5AA0007B7D7C7D7C7D7C7D7C7D7C7D7D7E28
:00000001FF
```

rr_audio_fuel.hex

```
:020000040000FA
:200000009AC2B0AC936A53503A92C7ABAE8E6B4F523691C6ABAB8962524A42B2BDADA58015
:2000200059543F5CC6B1B1987651563682C7ADAC8D65514D3BA4BFABA27B5555387CC7AEFA
:20004000AA9162564451B9B9ABA072564F3D95C5AAA9835A553B74C7AEAB9161564352B9C3
:20006000B8A89B6559454BA9BFA6A26C584B3F9AC5A5A66E594C409DC4A6A56F594D40A09B
:20008000C2A7A36E584C40A1C1A6A069584847A9BCA6995F5A3F5FC0B1A88D5B563B85C533
:2000A000A9A77958503E94C2A69F66584452B7B4A78A5A553D90C4A6A56C5B4456BBB4A704
:2000C0008B59573981C4A6A2685A4160C3ADA97C5B4D48AEB9A7945D573D86C3A6A1695952
:2000E0004168C2ABA7745A455CC1AEA97D5A4D47B1B6A68958533FA0BBA58C58553EA5BAFB
:20010000A68F5C5244ACB8A78D5A5243A6B8A68A585241A7B6A682584C50B8AFA7775A441C
:2001200068C5A7A66A5B3D78C4A59B5D593B91BCA68B5A4E4BB7AFA57259427AC6A49D5C70
:20014000593FA4B6A7825A4956BDA9A164593E9DB7A87E5D446DC4A79D5F583EAAB3A87689
:200160005C3F6DBCAA8D614959BBAE9F69554194BAA278594178BFA7895F4468BCAD9168A4
:20018000465BB3B595714952A6BB98784C4A96C0997A514684C29B7D514993BEA07D5543B0
:2001A0008CBF9F7B54468EBDA07C54448CC09A794E4D9FBD96754957AAB8926E475CAEB4D9
:2001C0008E69436BB7AC875F4385BEA17D524B9FBC96734956ACB58C674365B7A981594671
:2001E0009DBB99714B50A8B587654084BEA07C4E51A3B78C684368BAA47953499FB88E6AC1
:200200004175BDA17D4F50A9B589644180BE98734A57AFAD7F584A9EB78F674286BE9B76D9
:200220004959B3A97F5253ABB38B624287BD916C4281BF9974466AB6A57B504FABAE7F56EC
:200240004BA6B2855B48A0B689604599B98B644391BA8E664287BC8E694182BA91664695C1
:20026000B98D644595BA8B65418DB88B5E4CA7B3855B47A1B084525BB6A87C4F57B2A2744A
:20028000486CBC9870438DBA906449A4B1875655ACAA7A4E5EB69C6F457ABA8C614AA3AE5A
:2002A000824D68BA9D74448BBA8D6247A6AB7B4D61B6996848A0B3855555B49D6F4794B631
:2002C000885A4FB1A172468FB8885C4CADA273448AB7865364BC986C499FB0815063B99013
:2002E0006744A4A678447BB7865655B5976A449DAA79467FB488506EB9925F54B59F724469
:2003000097AF7F4972B58C516FBA925E52ADA2684EA0AA71498EB57A4E86B8805169B98B90
:200320005768C087819F785C584663BEAAAA9677595A4467BAABA894735759407CBFA6A8B8
:200340008C6B5655439DB9A6A38362584C52B4AEA89879585B4176BBA8A48B655A4D51AF1B
```

```
:20036000B2A69C785B594388BCA6A588635A4C55AEB1A3986C5C514B9BB9A2A1785D5744A0
:200380008CBEA4A484605A4668BAA9A38B605E446DBAABA38F625F4668B6ADA292645E4838
:2003A00062B3AEA291625F456EB9AAA38B605E4473BBA7A2845E5B4479BBA4A0775D56475F
:2003C0009AB8A09D6B5F4B5FB6ACA38E625D4581BDA3A3795F534FA7B4A198655F466BBAC2
:2003E000A7A17D5D584695B7A097635F4577BCA2A172604D5CB3ACA2895F5A468CB89F98BE
:20040000655E4772B9A29F6F5F4A6DBAA4A275604E5DB5A9A2805D564AA5AFA18760555162
:20042000ACAEA1895E5A479CB2A0875D5948A2B0A0835D574DABACA07D5E5354B1A7A0736C
:200440005F4D62B6A39C6B5F488BB99F97615E4599B1A0845F525FB8A4A171604970B8A0AC
:2004600096615D469EAFA07E5E4F63BAA09C64604499AFA180604F5FB6A29A655E479AADAF
:20048000A177624880B7A28E625650A8AA99705A4C9CB19E7D5F4C7CB5A28863515CACAA56
:2004A0008E6A5355A0B19272555092B69579584E83B8987E5A4E7CB69D805E4C8BB59F8154
:2004C000604A7BB4A07F604C8AB39F805F4B7CB59C7F5C4E8EB6987B575499B39578545875
:2004E0009CB09172515AA1AE8A6D4C70AFA584624D89B49A7C58549BB29173505DA6AB86A5
:20050000694A83B29D7B5A5196B28E734B73AFA581624B81B593755160ABA787644D8EB4EC
:2005200094784F69AEA586605095B291744C73B2A0815A559FAE8B6C4A7CB39875555CA8BE
:20054000A4825B57A2AD8B6A4B7DB492734D77B39C7D5658A2A87F5F52A0AB8763519AAF77
:200560008B694F8CB191704D74B09674506AAF9A775264AC9D795461AB9E7C536EB29B7B8A
:20058000526BB09C7A5363AD9C795175B396764E75B195724F7EB28E6E4B88AF8D67539C80
:2005A000AC8562519AA87E5A5CAA9E7B5172B393744D8DAF8C6654A3A5855867B09A7A4F33
:2005C0007FB18F6D4F97AA845E5AA99B765175AE8D6657A2A481576AB191714E86AB855C10
:2005E00066AF97784E7EAD876060AD9776507FAB88605DA799725390AA805C6DB08F715115
:2006000092A47E586DAC8A6955A199775186A7815B69AA8C69579D9A755590A27F5780AAD0
:20062000886266A8916F56969D765992A27E597DA8866168A38F6467A3946D5B939D6F5DF3
:2006400090A2775E8BA37B5E78AC989C897565655889A8989A87756467568BA59997856F40
:2006600067625F9BA09893806B695D6EA599998D7A686A5A82A49696867368665E939E9586
:20068000907F6B6B5E7FA2959487736B6667969A938E7A6D6B617F9E9291816F6E63799C81
:2006A000949188736F666E9796908A76706B678A9A8E8E78716C688A998D8E7A726D698B50
:2006C000978D8D7B736E6A8B968D8C7A736F6C8B948C8A79736F6D8A938A8977756D779212
:2006E0008E8B8476746D7A918C8A7F75746D839189887A7671738C8D89837776707E8F8800
```

```
:20070000887C7773778C8B8883787772808D87877B79747A8C888780797774848B86847AE2
:2007200079747D8A86857C7976798986857F7A7878868785807B7978868684817B7A7783DB
:200740008784827C7B79848683827C7B79818683817C7C79818583817C7C7A828482807CA1
:200760007C7B8383827F7D7C7C8382827E7D7C7F8482817E7D7C808382807E7D7D82828189
:200780007F7E7D7E8281817E7E7D808281807E7E7E8281817F7E7D808181807F7E7E81816A
:2007A000807F7F7E8081807F7F7E7F8180807F7F7F8080807F7F7F808080807F7F7F808049
:2007C000807F7F7F8080807F7F7F8080807F7F7F8080807F7F7F8080807F7F7F8080808028
:2007E0007F7F7F8080807F7F8080807F7F7F8080807F7F7F8080807F807F8080807F807F08
:00000001FF
```

audioROM_explode.v

```
// megafunction wizard: %ROM: 1-PORT%
// GENERATION: STANDARD
// VERSION: WM1.0
// MODULE: altsyncram

// ============================================================
// File Name: audioROM_explode.v
// Megafunction Name(s):
//                      altsyncram
//
// Simulation Library Files(s):
//                      altera_mf
// ============================================================
// ************************************************************
// THIS IS A WIZARD-GENERATED FILE. DO NOT EDIT THIS FILE!
//
// 21.1.0 Build 842 10/21/2021 SJ Lite Edition
// ************************************************************


//Copyright (C) 2021  Intel Corporation. All rights reserved.
//Your use of Intel Corporation's design tools, logic functions
//and other software and tools, and any partner logic
//functions, and any output files from any of the foregoing
//(including device programming or simulation files), and any
//associated documentation or information are expressly subject
//to the terms and conditions of the Intel Program License
//Subscription Agreement, the Intel Quartus Prime License Agreement,
//the Intel FPGA IP License Agreement, or other applicable license
//agreement, including, without limitation, that your use is for
//the sole purpose of programming logic devices manufactured by
//Intel and sold by Intel or its authorized distributors.  Please
//refer to the applicable agreement for further details, at
//https://fpgasoftware.intel.com/eula.


// synopsys translate_off
`timescale 1 ps / 1 ps
// synopsys translate_on
module audioROM_explode (
	address,
	clock,
	q);
	input	[11:0]  address;
	input	  clock;
	output	[7:0]  q;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_off
`endif
	tri1	  clock;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_on
`endif

	wire [7:0] sub_wire0;
	wire [7:0] q = sub_wire0[7:0];

	altsyncram	altsyncram_component (
				.address_a (address),
				.clock0 (clock),
				.q_a (sub_wire0),
				.aclr0 (1'b0),
				.aclr1 (1'b0),
				.address_b (1'b1),
				.addressstall_a (1'b0),
				.addressstall_b (1'b0),
				.byteena_a (1'b1),
				.byteena_b (1'b1),
				.clock1 (1'b1),
				.clocken0 (1'b1),
				.clocken1 (1'b1),
                                        .clocken2 (1'b1),
                                        .clocken3 (1'b1),
                                        .data_a ({8{1'b1}}),
                                        .data_b (1'b1),
                                        .eccstatus (),
                                        .q_b (),
                                        .rden_a (1'b1),
                                        .rden_b (1'b1),
                                        .wren_a (1'b0),
                                        .wren_b (1'b0));
          defparam
                    altsyncram_component.address_aclr_a = "NONE",
                    altsyncram_component.clock_enable_input_a = "BYPASS",
                    altsyncram_component.clock_enable_output_a = "BYPASS",
`ifdef NO_PLI
                    altsyncram_component.init_file = "rr_audio_explode.rif"
`else
                    altsyncram_component.init_file = "rr_audio_explode.hex"
`endif
,
                    altsyncram_component.intended_device_family = "Cyclone V",
                    altsyncram_component.lpm_hint = "ENABLE_RUNTIME_MOD=NO",
                    altsyncram_component.lpm_type = "altsyncram",
                    altsyncram_component.numwords_a = 4096,
                    altsyncram_component.operation_mode = "ROM",
                    altsyncram_component.outdata_aclr_a = "NONE",
                    altsyncram_component.outdata_reg_a = "UNREGISTERED",
                    altsyncram_component.widthad_a = 12,
                    altsyncram_component.width_a = 8,
		altsyncram_component.width_byteena_a = 1;



endmodule


// ============================================================
// CNX file retrieval info
// ============================================================
// Retrieval info: PRIVATE: ADDRESSSTALL_A NUMERIC "0"
// Retrieval info: PRIVATE: AclrAddr NUMERIC "0"
// Retrieval info: PRIVATE: AclrByte NUMERIC "0"
// Retrieval info: PRIVATE: AclrOutput NUMERIC "0"
// Retrieval info: PRIVATE: BYTE_ENABLE NUMERIC "0"
// Retrieval info: PRIVATE: BYTE_SIZE NUMERIC "8"
// Retrieval info: PRIVATE: BlankMemory NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_INPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_OUTPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: Clken NUMERIC "0"
// Retrieval info: PRIVATE: IMPLEMENT_IN_LES NUMERIC "0"
// Retrieval info: PRIVATE: INIT_FILE_LAYOUT STRING "PORT_A"
// Retrieval info: PRIVATE: INIT_TO_SIM_X NUMERIC "0"
// Retrieval info: PRIVATE: INTENDED_DEVICE_FAMILY STRING "Cyclone V"
// Retrieval info: PRIVATE: JTAG_ENABLED NUMERIC "0"
// Retrieval info: PRIVATE: JTAG_ID STRING "NONE"
// Retrieval info: PRIVATE: MAXIMUM_DEPTH NUMERIC "0"
// Retrieval info: PRIVATE: MIFfilename STRING "rr_audio_explode.hex"
// Retrieval info: PRIVATE: NUMWORDS_A NUMERIC "4096"
// Retrieval info: PRIVATE: RAM_BLOCK_TYPE NUMERIC "0"
// Retrieval info: PRIVATE: RegAddr NUMERIC "1"
// Retrieval info: PRIVATE: RegOutput NUMERIC "0"
// Retrieval info: PRIVATE: SYNTH_WRAPPER_GEN_POSTFIX STRING "0"
// Retrieval info: PRIVATE: SingleClock NUMERIC "1"
// Retrieval info: PRIVATE: UseDQRAM NUMERIC "0"
// Retrieval info: PRIVATE: WidthAddr NUMERIC "12"
// Retrieval info: PRIVATE: WidthData NUMERIC "8"
// Retrieval info: PRIVATE: rden NUMERIC "0"
// Retrieval info: LIBRARY: altera_mf altera_mf.altera_mf_components.all
// Retrieval info: CONSTANT: ADDRESS_ACLR_A STRING "NONE"
// Retrieval info: CONSTANT: CLOCK_ENABLE_INPUT_A STRING "BYPASS"
// Retrieval info: CONSTANT: CLOCK_ENABLE_OUTPUT_A STRING "BYPASS"
// Retrieval info: CONSTANT: INIT_FILE STRING "rr_audio_explode.hex"
// Retrieval info: CONSTANT: INTENDED_DEVICE_FAMILY STRING "Cyclone V"
// Retrieval info: CONSTANT: LPM_HINT STRING "ENABLE_RUNTIME_MOD=NO"
// Retrieval info: CONSTANT: LPM_TYPE STRING "altsyncram"
// Retrieval info: CONSTANT: NUMWORDS_A NUMERIC "4096"
// Retrieval info: CONSTANT: OPERATION_MODE STRING "ROM"
// Retrieval info: CONSTANT: OUTDATA_ACLR_A STRING "NONE"
// Retrieval info: CONSTANT: OUTDATA_REG_A STRING "UNREGISTERED"
// Retrieval info: CONSTANT: WIDTHAD_A NUMERIC "12"
// Retrieval info: CONSTANT: WIDTH_A NUMERIC "8"
// Retrieval info: CONSTANT: WIDTH_BYTEENA_A NUMERIC "1"
// Retrieval info: USED_PORT: address 0 0 12 0 INPUT NODEFVAL "address[11..0]"
// Retrieval info: USED_PORT: clock 0 0 0 0 INPUT VCC "clock"
// Retrieval info: USED_PORT: q 0 0 8 0 OUTPUT NODEFVAL "q[7..0]"
// Retrieval info: CONNECT: @address_a 0 0 12 0 address 0 0 12 0
// Retrieval info: CONNECT: @clock0 0 0 0 0 clock 0 0 0 0
// Retrieval info: CONNECT: q 0 0 8 0 @q_a 0 0 8 0
// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_explode.v TRUE
// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_explode.inc FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_explode.cmp FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_explode.bsf FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_explode_inst.v FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL audioROM_explode_bb.v TRUE
// Retrieval info: LIB_FILE: altera_mf
```

spriteROM.v

```
// megafunction wizard: %ROM: 1-PORT%
// GENERATION: STANDARD
// VERSION: WM1.0
// MODULE: altsyncram

// ============================================================
// File Name: spriteROM.v
// Megafunction Name(s):
//                         altsyncram
//
// Simulation Library Files(s):
//                         altera_mf
// ============================================================
// ************************************************************
// THIS IS A WIZARD-GENERATED FILE. DO NOT EDIT THIS FILE!
//
// 21.1.0 Build 842 10/21/2021 SJ Lite Edition
// ************************************************************



//Copyright (C) 2021  Intel Corporation. All rights reserved.
//Your use of Intel Corporation's design tools, logic functions
//and other software and tools, and any partner logic
//functions, and any output files from any of the foregoing
//(including device programming or simulation files), and any
//associated documentation or information are expressly subject
//to the terms and conditions of the Intel Program License
//Subscription Agreement, the Intel Quartus Prime License Agreement,
//the Intel FPGA IP License Agreement, or other applicable license
//agreement, including, without limitation, that your use is for
//the sole purpose of programming logic devices manufactured by
//Intel and sold by Intel or its authorized distributors.  Please
//refer to the applicable agreement for further details, at
//https://fpgasoftware.intel.com/eula.



// synopsys translate_off
`timescale 1 ps / 1 ps
// synopsys translate_on
module spriteROM (
	address,
	clock,
	q);

	input	[12:0]  address;
	input	  clock;
	output	[5:0]	q;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_off
`endif
	tri1	clock;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_on
`endif

	wire [5:0] sub_wire0;
	wire [5:0] q = sub_wire0[5:0];

	altsyncram	altsyncram_component (
				.address_a (address),
				.clock0 (clock),
				.q_a (sub_wire0),
				.aclr0 (1'b0),
				.aclr1 (1'b0),
				.address_b (1'b1),
				.addressstall_a (1'b0),
				.addressstall_b (1'b0),
				.byteena_a (1'b1),
				.byteena_b (1'b1),
				.clock1 (1'b1),
				.clocken0 (1'b1),
				.clocken1 (1'b1),
				.clocken2 (1'b1),
				.clocken3 (1'b1),
				.data_a ({6{1'b1}}),
                .data_b (1'b1),
                .eccstatus (),
                .q_b (),
                .rden_a (1'b1),
                .rden_b (1'b1),
                .wren_a (1'b0),
                .wren_b (1'b0));
    defparam
        altsyncram_component.address_aclr_a = "NONE",
        altsyncram_component.clock_enable_input_a = "BYPASS",
        altsyncram_component.clock_enable_output_a = "BYPASS",
        altsyncram_component.init_file = "mif/sprites.mif",
        altsyncram_component.intended_device_family = "Cyclone V",
        altsyncram_component.lpm_hint = "ENABLE_RUNTIME_MOD=NO",
        altsyncram_component.lpm_type = "altsyncram",
        altsyncram_component.numwords_a = 8192,
        altsyncram_component.operation_mode = "ROM",
        altsyncram_component.outdata_aclr_a = "NONE",
        altsyncram_component.outdata_reg_a = "UNREGISTERED",
        altsyncram_component.widthad_a = 13,
        altsyncram_component.width_a = 6,
        altsyncram_component.width_byteena_a = 1;


endmodule

// ============================================================
// CNX file retrieval info
// ============================================================
// Retrieval info: PRIVATE: ADDRESSSTALL_A NUMERIC "0"
// Retrieval info: PRIVATE: AclrAddr NUMERIC "0"
// Retrieval info: PRIVATE: AclrByte NUMERIC "0"
// Retrieval info: PRIVATE: AclrOutput NUMERIC "0"
// Retrieval info: PRIVATE: BYTE_ENABLE NUMERIC "0"
// Retrieval info: PRIVATE: BYTE_SIZE NUMERIC "8"
// Retrieval info: PRIVATE: BlankMemory NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_INPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_OUTPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: Clken NUMERIC "0"
// Retrieval info: PRIVATE: IMPLEMENT_IN_LES NUMERIC "0"
// Retrieval info: PRIVATE: INIT_FILE_LAYOUT STRING "PORT_A"
// Retrieval info: PRIVATE: INIT_TO_SIM_X NUMERIC "0"
// Retrieval info: PRIVATE: INTENDED_DEVICE_FAMILY STRING "Cyclone V"
// Retrieval info: PRIVATE: JTAG_ENABLED NUMERIC "0"
// Retrieval info: PRIVATE: JTAG_ID STRING "NONE"
// Retrieval info: PRIVATE: MAXIMUM_DEPTH NUMERIC "0"
// Retrieval info: PRIVATE: MIFfilename STRING "./river_raid_hw/mif/sprites.mif"
// Retrieval info: PRIVATE: NUMWORDS_A NUMERIC "8192"
// Retrieval info: PRIVATE: RAM_BLOCK_TYPE NUMERIC "0"
// Retrieval info: PRIVATE: RegAddr NUMERIC "1"
// Retrieval info: PRIVATE: RegOutput NUMERIC "0"
// Retrieval info: PRIVATE: SYNTH_WRAPPER_GEN_POSTFIX STRING "0"
// Retrieval info: PRIVATE: SingleClock NUMERIC "1"
// Retrieval info: PRIVATE: UseDQRAM NUMERIC "0"
// Retrieval info: PRIVATE: WidthAddr NUMERIC "13"
// Retrieval info: PRIVATE: WidthData NUMERIC "6"
// Retrieval info: PRIVATE: rden NUMERIC "0"
// Retrieval info: LIBRARY: altera_mf altera_mf.altera_mf_components.all
// Retrieval info: CONSTANT: ADDRESS_ACLR_A STRING "NONE"
// Retrieval info: CONSTANT: CLOCK_ENABLE_INPUT_A STRING "BYPASS"
// Retrieval info: CONSTANT: CLOCK_ENABLE_OUTPUT_A STRING "BYPASS"
// Retrieval info: CONSTANT: INIT_FILE STRING "./river_raid_hw/mif/sprites.mif"
// Retrieval info: CONSTANT: INTENDED_DEVICE_FAMILY STRING "Cyclone V"
// Retrieval info: CONSTANT: LPM_HINT STRING "ENABLE_RUNTIME_MOD=NO"
// Retrieval info: CONSTANT: LPM_TYPE STRING "altsyncram"
// Retrieval info: CONSTANT: NUMWORDS_A NUMERIC "8192"
// Retrieval info: CONSTANT: OPERATION_MODE STRING "ROM"
// Retrieval info: CONSTANT: OUTDATA_ACLR_A STRING "NONE"
// Retrieval info: CONSTANT: OUTDATA_REG_A STRING "UNREGISTERED"
// Retrieval info: CONSTANT: WIDTHAD_A NUMERIC "13"
// Retrieval info: CONSTANT: WIDTH_A NUMERIC "6"
// Retrieval info: CONSTANT: WIDTH_BYTEENA_A NUMERIC "1"
// Retrieval info: USED_PORT: address 0 0 13 0 INPUT NODEFVAL "address[12..0]"
// Retrieval info: USED_PORT: clock 0 0 0 0 INPUT VCC "clock"
// Retrieval info: USED_PORT: q 0 0 6 0 OUTPUT NODEFVAL "q[5..0]"
// Retrieval info: CONNECT: @address_a 0 0 13 0 address 0 0 13 0
// Retrieval info: CONNECT: @clock0 0 0 0 0 clock 0 0 0 0
// Retrieval info: CONNECT: q 0 0 6 0 @q_a 0 0 6 0
// Retrieval info: GEN_FILE: TYPE_NORMAL spriteROM.v TRUE
// Retrieval info: GEN_FILE: TYPE_NORMAL spriteROM.inc FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL spriteROM.cmp FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL spriteROM.bsf FALSE
// Retrieval info: GEN_FILE: TYPE_NORMAL spriteROM_inst.v TRUE
// Retrieval info: GEN_FILE: TYPE_NORMAL spriteROM_bb.v TRUE
// Retrieval info: LIB_FILE: altera_mf
```

background.mif

```
Depth = 8192;

Width = 5;

Address_radix=dec;

Data_radix=bin;

Content

BEGIN

        [0..8191] : 00000;

        8176 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

        8160 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

        8144 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

        8128 : 00111 00111 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 01100 00111 00111 00111;

        8112 : 00011 00011 00111 00111 00111 11100 11100 01100 01100 00111 00111 01100 00111 10011 00011 10011;

        8096 : 00011 00011 00011 00011 00011 00111 00111 00111 00111 10011 00011 00111 00011 00011 00011 00011;

        8080 : 11011 11011 00011 00011 00011 00011 10011 00011 00011 00011 00011 00011 00011 11011 11011 11011;

        8064 : 00011 11011 11011 11011 00011 11011 11011 00011 11011 11011 11011 00011 11011 11011 11011 11011;

        8048 : 11010 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11010 11011 11011;

        8032 : 11010 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

        8016 : 11011 11010 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;
```

```
	8000 : 11011 11011 11011 11011 11011 11011 11011 00011 11011 11010 11011 11011 11011 11011 11011 11011;

	7984 : 11011 11011 11011 11011 11011 11011 11011 11010 11011 11010 11011 11011 11011 11011 11011 11011;

	7968 : 11011 11011 11011 11011 11011 11011 00011 11010 11011 11011 11011 11011 11011 11011 11011 11011;

	7952 : 11011 11011 11011 11011 11011 11011 11010 11011 11011 11011 11011 11010 11011 11011 11011 11011;

	7936 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	7920 : 11100 11100 01100 11100 01100 01100 01100 01100 01100 01100 01100 01100 11100 01100 11100 11100;

	7904 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	7888 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	7872 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	7856 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	7840 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	7824 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	7808 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	7792 : 11100 11100 11100 11100 11100 01100 01100 01100 01100 01100 01100 11100 11100 11100 11100 11100;

	7776 : 11100 11100 11100 01100 01100 11100 11100 11100 11100 11100 01100 01100 01100 11100 11100 11100;

	7760 : 11100 01100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 01100 11100;

	7744 : 01100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 01100 01100;
```

```
        7728 : 11100 11100 01100 01100 01100 11100 01100 01100 01100 01100 01100 01100 01100
01100 01100 01100;

        7712 : 01100 01100 00111 01100 01100 01100 01100 00111 01100 00111 00111 00111 01100
01100 01100 00111;

        7696 : 00111 00111 01100 01100 00111 01100 00111 00111 00111 00111 01100 00111 00111
00111 00111 00111;

        7680 : 00111 00111 00111 00111 00111 00111 00111 00111 00111 00111 00111 00111 00111
00111 00111 00111;

        7664 : 10010 10010 10010 10010 10010 10010 10010 10010 10010 10010 10011 10011 10011
10011 11100 11100;

        7648 : 10101 10101 10101 10101 10101 10101 10101 10101 10101 10101 10010 10010 10011
10011 10011 11100;

        7632 : 00101 00101 00101 10101 10101 10101 10101 10101 10101 10101 10101 10101 10010
10011 10011 10011;

        7616 : 10101 10101 10101 00101 00101 00101 00101 10101 10101 10101 10101 10110 10010
10011 10011 10011;

        7600 : 10101 10101 10101 10101 10101 10101 10101 10101 00101 00101 10101 10110 10110
10010 10011 10011;

        7584 : 10101 10101 10101 00101 00101 00101 00101 00101 10101 10101 10101 10111 10110
10010 10011 10011;

        7568 : 00101 00101 00101 00101 00101 00101 00101 00101 00101 10101 10110 10111 10110
10010 11100 11100;

        7552 : 10101 10101 10101 10101 10101 10101 10101 00101 00101 00101 10111 10111 10111
10010 11100 11100;

        7536 : 00101 00101 00101 00101 00101 00101 00101 00101 00101 00101 10110 10111 10111
10101 11100 11100;

        7520 : 10110 10110 00101 00101 10110 10110 10110 00101 00101 00101 10111 10111 10111
10101 11100 11100;

        7504 : 10110 10110 10110 10110 10110 10110 00101 10110 10110 10110 10111 10111 10111
10101 11100 11100;

        7488 : 00101 00101 10110 10110 10110 10110 10110 10110 10110 10110 10110 10111 10110
10101 11100 11100;

        7472 : 10110 10110 00101 10110 10110 10110 10110 00101 00101 00101 00101 10110 10101
11100 11100 11100;
```

```
        7456 : 10110 10110 10110 00101 00101 00101 00101 10110 10110 10110 10110 10110 10101 11100 11100 11100;

        7440 : 10101 10101 10101 10110 10110 10110 10110 10110 10110 00101 10101 10101 11100 11100 11100 11100;

        7424 : 11100 11100 11100 10101 10101 10101 10101 10101 10101 10101 11100 11100 11100 11100 11100 11100;

        7408 : 10011 10011 10011 10011 10011 10011 10011 10010 10010 10010 10010 10010 10010 10010 10010 10010;

        7392 : 10010 10010 10010 10010 10010 10010 10010 10101 10101 10101 10101 10101 10101 10101 10101 10101;

        7376 : 10101 10101 10101 10101 10101 10101 10101 10101 10101 10101 10101 10101 10101 10101 10101 00101;

        7360 : 10101 10101 10101 10101 10101 10101 10101 10101 10101 10101 10101 00101 00101 00101 00101 10101;

        7344 : 00101 00101 00101 00101 10101 10101 10101 10101 10101 00101 00101 10101 10101 10101 10101 10101;

        7328 : 00101 00101 10101 10101 10101 10101 10101 00101 00101 00101 00101 00101 00101 00101 10101 10101;

        7312 : 00101 00101 00101 00101 00101 00101 00101 10101 10101 10101 10101 10101 00101 00101 00101 00101;

        7296 : 10101 10101 00101 00101 00101 10101 10101 00101 00101 00101 00101 00101 10101 10101 10101 10101;

        7280 : 00101 00101 10101 10101 10101 00101 00101 00101 00101 00101 10110 00101 00101 00101 00101 00101;

        7264 : 00101 00101 10110 10110 10110 10101 10101 10110 10110 10110 10110 10110 10110 10110 10110 10110;

        7248 : 10110 10110 10110 10110 10110 10110 10110 00101 00101 00101 10110 10110 00101 00101 00101 00101;

        7232 : 10110 10110 10110 00101 00101 00101 00101 10110 10110 10110 00101 00101 10110 10110 10110 10110;

        7216 : 10110 00101 00101 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110;

        7200 : 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110;
```

7184 : 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10101 10101 10101 10101 10101 10101;

7168 : 10101 10101 10101 10101 10101 10101 10101 10101 10101 10101 11100 11100 11100 11100 11100 11100;

7152 : 11100 11100 11100 11100 11100 11100 10010 10010 10010 10010 10010 10010 10010 10011 10011 10011;

7136 : 11100 11100 11100 11100 10010 10010 10101 10101 10101 10101 10101 10101 10101 10010 10010 10010;

7120 : 11100 11100 11100 10010 10101 10101 10101 10101 10101 10101 10101 10101 10101 10101 10101 10101;

7104 : 11100 11100 11100 10010 10110 10101 10101 10101 10101 00101 00101 00101 00101 10101 10101 10101;

7088 : 11100 11100 10010 10110 10110 10101 00101 00101 10101 10101 10101 10101 10101 10101 10101 00101;

7072 : 11100 11100 10010 10110 10111 10101 10101 10101 00101 00101 00101 00101 00101 00101 00101 00101;

7056 : 11100 11100 10010 10110 10111 10110 10101 00101 00101 00101 00101 00101 00101 00101 00101 00101;

7040 : 11100 11100 10010 10111 10111 10111 00101 00101 00101 10101 10101 10101 10101 00101 00101 00101;

7024 : 11100 11100 10101 10111 10111 10110 00101 00101 00101 00101 00101 00101 00101 10101 10101 10101;

7008 : 11100 11100 10101 10111 10111 10111 00101 00101 00101 10110 10110 10110 00101 00101 00101 00101;

6992 : 11100 11100 10101 10111 10111 10111 10110 10110 10110 00101 10110 10110 10110 10110 00101 10110;

6976 : 11100 11100 10101 10110 10111 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110;

6960 : 11100 11100 11100 10101 10110 00101 00101 00101 00101 10110 10110 10110 10110 10110 10110 10110;

6944 : 11100 11100 11100 10101 10110 10110 10110 10110 10110 00101 00101 00101 00101 10110 10110 10110;

6928 : 11100 11100 11100 11100 10101 10101 00101 10110 10110 10110 10110 10110 10110 10110 10110 10110;

```
	6912 : 11100 11100 11100 11100 11100 11100 10101 10101 10101 10101 10101 10101 10101 10101 10101 10101;

	6896 : 11011 11011 11011 11011 11011 11011 11011 00011 11011 11011 11011 11011 11011 11011 11011 11011;

	6880 : 11011 11011 11011 11011 00011 00011 00011 00111 00011 11011 11011 11011 11011 11011 11011 11011;

	6864 : 11011 11011 11011 00011 11010 00011 00111 00011 00111 11011 11011 11011 00011 11011 11011 11011;

	6848 : 11011 11011 11011 11010 11010 11010 11010 00111 11010 11011 11010 11011 11010 11011 11011 11011;

	6832 : 11011 11011 11011 11010 11011 11010 11010 11011 11010 11011 11011 11011 11010 00011 11011 11011;

	6816 : 11011 11010 11011 11011 11011 11010 11011 11011 11011 11011 11011 11011 11011 11010 11011 11011;

	6800 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	6784 : 11011 11011 00011 00011 11011 11011 11011 11011 11011 00011 00011 00011 00011 11011 11011 11011;

	6768 : 11011 00011 00111 00111 00011 11011 11011 11011 00011 11010 11010 00011 11010 00011 11011 11011;

	6752 : 00011 01100 00111 00111 01100 00011 11011 11011 11010 11010 11010 11010 00011 00111 00011 11011;

	6736 : 00111 00111 00011 11010 00111 00111 11011 11011 11010 11011 11010 11010 00111 00011 00111 11011;

	6720 : 00111 00111 00011 00011 00111 00111 11011 11011 11011 11011 11010 11011 11011 00111 11011 11011;

	6704 : 11010 01100 00111 00111 01100 11010 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	6688 : 11011 11011 00111 00111 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	6672 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	6656 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;
	6640 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	6624 : 11100 11100 01100 01100 01100 01100 01100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	6608 : 11100 01100 01100 00011 00011 00011 01100 01100 11100 11100 11100 11100 11100 11100 11100 11100;

	6592 : 01100 01100 00011 11011 11011 11011 00011 01100 01100 11100 11100 11100 11100 11100 11100 11100;

	6576 : 01100 00011 11011 00011 11010 11011 11011 00011 01100 11100 11100 11100 11100 11100 11100 11100;

	6560 : 01100 00011 11011 11010 00011 00011 11011 00011 01100 11100 11100 11100 11100 11100 11100 11100;

	6544 : 01100 00011 11011 11011 00011 01100 00011 00011 01100 11100 11100 11100 11100 11100 11100 11100;

	6528 : 01100 01100 00011 11011 11011 00011 01100 01100 01100 01100 01100 01100 01100 01100 11100 11100;

	6512 : 11100 01100 01100 00011 00011 00011 01100 01100 01100 01100 00011 00011 00011 01100 01100 11100;

	6496 : 11100 11100 01100 01100 01100 01100 01100 01100 01100 00011 11011 11011 00011 01100 01100 01100;

	6480 : 11100 11100 11100 11100 11100 11100 11100 01100 00011 11011 11011 00011 01100 00011 00011 01100;

	6464 : 11100 11100 11100 11100 11100 11100 11100 01100 00011 11011 11010 00011 00011 11011 00011 01100;

	6448 : 11100 11100 11100 11100 11100 11100 11100 01100 00011 11011 00011 11010 11011 11011 00011 01100;

	6432 : 11100 11100 11100 11100 11100 11100 11100 01100 01100 00011 11011 11011 11011 00011 01100 01100;

	6416 : 11100 11100 11100 11100 11100 11100 11100 11100 01100 01100 00011 00011 00011 01100 01100 11100;

	6400 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 01100 01100 01100 01100 01100 11100 11100;

	6384 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;
```

```
6368 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

6352 : 11100 11100 11100 11100 01100 01100 01100 01100 01100 01100 01100 11100 11100 11100 11100 11100;

6336 : 11100 11100 11100 01100 01100 00011 00011 00011 00011 00011 01100 01100 11100 11100 11100 11100;

6320 : 11100 11100 01100 01100 00011 11011 11011 11011 11011 11011 00011 01100 01100 11100 11100 11100;

6304 : 11100 11100 01100 00011 11011 11011 11010 11011 11010 11011 11011 00011 01100 11100 11100 11100;

6288 : 11100 11100 01100 00011 11011 11011 11011 11010 00011 11011 11011 00011 01100 11100 11100 11100;

6272 : 11100 11100 01100 00011 11011 11011 11011 00011 11010 11011 11011 00011 01100 11100 11100 11100;

6256 : 11100 11100 01100 00011 00011 00011 00011 11010 11011 11010 11011 00011 01100 11100 11100 11100;

6240 : 11100 11100 01100 00011 01100 01100 00011 11010 11011 11011 11011 00011 01100 11100 11100 11100;

6224 : 11100 11100 01100 01100 01100 01100 00011 11011 11011 11011 00011 01100 01100 11100 11100 11100;

6208 : 11100 11100 11100 01100 01100 00011 00011 00011 00011 00011 01100 01100 11100 11100 11100 11100;

6192 : 11100 11100 11100 11100 01100 01100 01100 01100 01100 01100 01100 11100 11100 11100 11100 11100;

6176 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

6160 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

6144 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

6128 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

6112 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;
```

```
	6096 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	6080 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	6064 : 11100 11100 11100 01100 01100 01100 11100 11100 11100 11100 01100 01100 01100 11100 11100 11100;

	6048 : 11100 11100 11100 11100 11100 01100 01100 00111 00111 01100 01100 11100 11100 11100 11100 11100;

	6032 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	6016 : 11100 11100 01100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 01100 11100 11100;

	6000 : 11100 11100 11100 01100 00111 01100 01100 11100 11100 01100 01100 00111 01100 11100 11100 11100;

	5984 : 11100 11100 11100 11100 01100 01100 00111 00111 00111 00111 01100 01100 11100 11100 11100 11100;

	5968 : 11100 11100 11100 11100 11100 11100 11100 01100 01100 11100 11100 11100 11100 11100 11100 11100;

	5952 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	5936 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	5920 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	5904 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	5888 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	5872 : 11011 11011 11011 11011 11011 11011 11011 11011 00011 11011 11011 11011 11011 11011 11011 11011;

	5856 : 11011 11011 11011 00011 00011 00011 00011 00011 00111 00011 11011 11011 11011 00011 11011 11011;

	5840 : 11011 11011 00011 00111 00111 00011 11010 00111 00011 00111 11011 11011 00011 00111 00011 11011;
```

```
5824 : 11011 00011 01100 00111 00111 01100 00011 11010 00111 11011 11010 11011 00111 00011 00111 11011;

5808 : 11011 00111 00111 11010 11011 00111 00111 11011 11010 11011 11011 11011 11010 00111 11011 11011;

5792 : 11011 00111 00111 00011 00011 00111 00111 11011 11011 00011 11011 11011 11011 11010 11011 11011;

5776 : 11011 11011 01100 00111 00111 01100 11011 11011 00011 00111 00011 11011 11011 11011 11011 11011;

5760 : 11011 11011 11011 00111 00111 11011 11011 11011 00111 00011 00111 00011 00011 11011 11011 11011;

5744 : 11011 11011 00011 11011 11011 11011 11011 11011 00011 00111 11010 00011 00011 00011 11011 11011;

5728 : 11011 00011 00111 00011 00011 11011 11011 11011 11010 11010 00011 00111 00111 00011 11011 11011;

5712 : 00011 00111 00011 00111 11010 00011 11011 11011 11010 00011 01100 00111 00111 01100 00011 11011;

5696 : 11010 11010 00111 11010 11010 11010 11011 11011 11011 00111 00111 11011 11011 00111 00111 11011;

5680 : 11010 11011 11010 11010 11011 11010 00011 11011 11011 00111 00111 00011 00011 00111 00111 11011;

5664 : 11011 11011 11010 11011 11011 00011 00111 00011 11011 11011 01100 00111 00111 01100 11011 11011;

5648 : 11011 11011 11011 11011 11011 00111 00011 00111 11011 11011 11011 00111 00111 11011 11011 11011;

5632 : 11011 11011 11011 11011 11011 11011 00111 11011 11011 11011 11011 11011 11011 11011 11011 11011;

5616 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

5600 : 11011 11011 11011 11011 11010 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

5584 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

5568 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11010 11011 11011 11011 11011;
```

```
	5552 : 11011 11011 11011 11011 11011 11011 00011 11011 00011 11011 11011 11011 11011 11011 11011 11011;

	5536 : 11011 11011 11011 11011 11011 11011 11010 11011 11010 11011 11011 11011 11011 11011 11011 11011;

	5520 : 11011 11011 11011 11011 11011 11011 11011 11011 11010 00011 11011 11011 11011 11011 11011 11011;

	5504 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11010 11011 11011 11011 11011 11011 11011;

	5488 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	5472 : 11011 00011 00011 00011 10011 00011 00011 00011 00011 00011 00011 00011 00011 00011 00011 11011;

	5456 : 00011 11010 00101 10011 00101 10011 00011 00011 00101 00011 00011 00011 00011 00011 00011 00011;

	5440 : 11010 00101 00101 00101 00101 00101 00101 00101 00101 00101 00101 00101 00101 00101 00101 00011;

	5424 : 11010 00101 11101 11101 11101 11101 11101 11101 11101 11101 11101 11101 11101 11101 00101 00011;

	5408 : 11011 00101 11101 11101 10101 10101 00101 00101 00101 00101 10101 10101 11101 11101 00101 00011;

	5392 : 11011 00101 11101 11101 11101 11101 10101 00101 00101 10101 11101 11101 11101 11101 00101 00011;

	5376 : 11011 00101 11101 10101 11101 11101 11101 10101 10101 11101 11101 11101 00101 11101 00101 00011;

	5360 : 11011 00101 11101 10101 00101 00101 11101 11101 11101 00101 00101 00101 00101 11101 00101 00011;

	5344 : 11011 00101 11101 10101 10101 10101 00101 11101 11101 11101 10101 10101 00101 11101 00101 00011;

	5328 : 11011 00101 11101 10101 11101 11101 11101 00101 00101 11101 11101 11101 10101 11101 00101 00011;

	5312 : 11011 00101 11101 11101 11101 11101 10101 10101 10101 10101 11101 11101 11101 11101 00101 11011;

	5296 : 11011 00101 11101 11101 11101 11101 11101 11101 11101 11101 11101 11101 11101 11101 00101 11011;
```

```
5280 : 11011 11101 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 11101 11011;

5264 : 11011 11101 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 11101 11011;

5248 : 11011 11101 10110 10111 11010 11010 11010 00100 11010 11010 10111 11010 11010 10110 11101 11011;

5232 : 11011 11101 10110 10111 10111 11010 11010 11010 10111 11010 11010 10111 11010 10110 11101 11011;

5216 : 11011 11101 10110 10111 11010 10111 10111 10111 10111 10111 11010 11010 11010 10110 11101 11011;

5200 : 00011 11101 10110 11010 11010 10111 11010 00100 10111 11010 11010 10111 11010 11010 11101 11011;

5184 : 11010 11101 11010 10111 11010 11010 00100 11010 10111 11010 11010 10111 10111 11010 11101 11011;

5168 : 11010 11101 10111 10111 10111 11010 00101 10111 10111 10111 11010 10111 11010 10110 11101 11011;

5152 : 11011 11101 10110 10111 11010 00101 00101 00101 10111 11010 00101 00101 00101 10110 11101 11011;

5136 : 11011 11101 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 10110 11101 11011;

5120 : 11011 11101 11101 11101 11101 11101 11101 11101 11101 11101 11101 11101 11101 11101 11101 11011;

5104 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

5088 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

5072 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

5056 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

5040 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

5024 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;
```

```
        5008 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

        4992 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

        4976 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

        4960 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

        4944 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

        4928 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

        4912 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

        4896 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

        4880 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

        4864 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

        4848 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        4832 : 11011 11011 11011 11011 00011 00011 10011 10011 00011 00011 00011 00011 00011
00011 11011 11011;

        4816 : 11011 11011 11011 10011 00101 10011 00101 00011 10011 00011 00011 00011 00011
00011 00011 00011;

        4800 : 11011 11011 10010 00011 10010 10010 10010 11011 10010 10010 10010 00011 10010
00011 00011 00011;

        4784 : 11011 10010 10010 11011 10101 10101 10101 10101 10010 10101 10101 11011 10101
10010 00011 00011;

        4768 : 00011 10010 10010 10010 10010 10101 00101 00101 10101 00101 00101 00101 00101
10101 11011 11011;

        4752 : 11011 10010 10101 10101 10101 10101 10101 00101 00101 00101 10110 10110 10110
10110 00101 11011;
```

```
4736 : 00011 11011 10101 10101 10101 00101 00101 00101 10110 10110 10110 10110 10110 10110 00101 11011;

4720 : 11010 10010 10010 10101 00101 00101 10110 10110 10110 10110 10110 10110 10110 10110 00101 11011;

4704 : 11010 00011 10010 00101 00101 10110 10110 10110 10110 10110 10110 10110 10110 10110 00101 11011;

4688 : 11011 11010 00101 10110 10110 10110 10110 10110 10110 10110 10110 00101 10110 10110 00101 11011;

4672 : 11011 11011 00101 10110 10110 10110 10110 10110 10110 10110 10110 00101 10110 00101 11011 11011;

4656 : 11011 11011 00101 10110 10110 10110 10110 10110 10110 00101 00101 10110 00101 11011 11011 11011;

4640 : 11011 11011 11011 00101 10110 10110 10110 00101 00101 00101 00101 00101 11011 11011 11011 11011;

4624 : 11011 11011 11011 11011 00101 00101 00101 11011 11011 11011 11011 11010 11011 11011 11011 11011;

4608 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

4592 : 11011 11011 11011 11011 11011 00011 00011 00011 00011 00011 00011 00011 00011 00011 00011 11011;

4576 : 11011 11011 11011 11011 00101 00011 00011 00011 00011 00011 00011 00011 00011 00011 00011 00011;

4560 : 11011 11011 11011 11011 11011 00011 10010 10010 10010 10010 00011 11011 11011 11011 11011 11011;

4544 : 11011 11011 11011 11011 11011 11011 10010 00100 00100 10010 11011 11010 11011 11011 11011 11011;

4528 : 11011 11011 11011 11011 11011 11011 10010 00100 00100 10010 11011 11011 11011 11011 11011 11011;

4512 : 11011 11011 11011 11011 11011 11011 10010 10101 10101 10010 11011 11011 11011 11011 11011 11011;

4496 : 10010 10010 10010 10010 10010 10010 10010 10010 10010 10010 10010 10010 10010 10010 10010 11011;

4480 : 10101 00100 00100 00100 00100 00100 00100 00100 00100 00100 00100 00100 00100 00100 00100 10010;
```

```
4464 : 00100 00100 10110 10110 00100 00100 00100 00100 00100 00100 00100 00100 00100 00100 00100 00100;

4448 : 11011 00100 00100 00100 10110 10110 00100 00100 10110 00100 00100 00100 10110 10110 00100 00100;

4432 : 00100 10110 10110 00100 10101 00100 00100 00100 00100 10110 00100 10110 00100 00100 00100 10101;

4416 : 10101 00100 00100 00100 00100 10101 00100 10110 10110 00100 00100 00100 00100 00100 00100 00100;

4400 : 00100 10110 10110 00100 00100 00100 00100 00100 00100 10101 00100 10101 00100 00100 10110 00100;

4384 : 00100 00100 00100 10101 00100 00100 00100 10101 10101 00100 00100 00100 00100 00100 00100 11011;

4368 : 00100 00100 10110 00100 00100 00100 00100 00100 00100 00100 00100 10110 10110 10110 10110 00100;

4352 : 11011 00100 00100 00100 00100 00100 00100 00100 00100 00100 00100 00100 00100 00100 00100 11011;

4336 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

4320 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

4304 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

4288 : 11100 11100 11100 11100 10011 10011 10011 10011 11100 11100 11100 11100 11100 11100 11100 11100;

4272 : 11100 11100 11100 10011 10011 10011 10011 10011 10011 11100 11100 11100 11100 11100 11100 11100;

4256 : 11100 11100 11100 11100 01100 00111 00111 01100 11100 11100 11100 11100 11100 11100 11100 11100;

4240 : 11100 11100 01100 00111 00111 10110 10110 00111 00111 01100 11100 11100 11100 11100 11100 11100;

4224 : 11100 11100 00111 00101 10110 10110 10110 10110 00101 00111 11100 11100 11100 11100 11100 11100;

4208 : 11100 11100 11100 01100 00101 10110 10110 00101 01100 11100 10011 10011 10011 11100 11100 11100;
```

```
        4192 : 11100 11100 11100 11100 11100 00101 00101 11100 10011 10011 10011 10011 10011 11100 11100 11100;

        4176 : 11100 11100 11100 11100 11100 11100 11100 11100 01100 00111 00111 01100 11100 10011 11100 11100;

        4160 : 11100 11100 11100 11100 11100 11100 01100 00111 00111 10110 10110 00111 00111 01100 11100 11100;

        4144 : 11100 11100 11100 11100 11100 11100 00111 00101 10110 10110 10110 10110 00101 00111 11100 11100;

        4128 : 11100 11100 11100 11100 11100 11100 11100 01100 00101 10110 10110 00101 01100 11100 11100 11100;

        4112 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 00101 00101 11100 11100 11100 11100 11100;

        4096 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

        4080 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

        4064 : 11100 11100 11100 11100 10011 10011 10011 10011 10011 10011 10011 11100 11100 11100 11100 11100;

        4048 : 11100 11100 10011 10011 10011 10011 10011 10011 10011 10011 10011 10011 10011 11100 11100 11100;

        4032 : 11100 10011 11100 01100 00111 00111 00111 00111 00111 00111 00111 00111 01100 11100 10011 11100;

        4016 : 11100 01100 00111 00111 10101 10101 00101 00101 10101 10101 10101 00101 00111 00111 01100 11100;

        4000 : 01100 00111 10101 10101 00101 00101 00101 00101 00101 10101 00101 00101 00101 10101 00111 01100;

        3984 : 00111 10101 10101 00101 00101 10110 10110 10110 10110 00101 10110 10110 00101 00101 00101 00111;

        3968 : 00111 10101 10101 00101 10110 10110 10110 10110 10110 00101 10110 10110 10110 00101 00101 01100;

        3952 : 01100 00111 10101 10110 10110 10110 10110 10110 10110 00101 10110 10110 10110 00101 01100 11100;

        3936 : 11100 01100 00101 10110 10110 10110 10110 10110 10110 10110 00101 10110 00101 00101 11100 11100;
```

```
3920 : 11100 11100 00101 10110 10110 10110 10110 10110 10110 10110 10110 00101 00101 11100 11100 11100;

3904 : 11100 11100 11100 00101 10110 10110 10110 10110 10110 10110 00101 00101 11100 11100 11100 11100;

3888 : 11100 11100 11100 11100 00101 00101 00101 00101 00101 00101 11100 11100 11100 11100 11100 11100;

3872 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

3856 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

3840 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

3824 : 11100 11100 01100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 01100 11100 11100;

3808 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

3792 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

3776 : 11100 11100 11100 11100 11100 11100 01100 01100 01100 01100 01100 11100 11100 11100 11100 11100;

3760 : 11100 11100 11100 01100 01100 11100 11100 11100 11100 11100 11100 01100 01100 11100 11100 11100;

3744 : 11100 11100 01100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 01100 11100 11100;

3728 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

3712 : 11100 01100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 01100 11100;

3696 : 11100 01100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 01100 11100;

3680 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

3664 : 11100 11100 01100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 01100 11100 11100;
```

```
        3648 : 11100 11100 11100 01100 01100 11100 11100 11100 11100 11100 11100 01100 01100
11100 11100 11100;

        3632 : 11100 11100 11100 11100 11100 11100 01100 01100 01100 01100 01100 11100 11100
11100 11100 11100;

        3616 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

        3600 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

        3584 : 11100 11100 01100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
01100 11100 11100;

        3568 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        3552 : 11011 11011 11011 11011 11010 00011 00011 00011 00011 00011 00011 00011 00011
00011 11011 11011;

        3536 : 11011 11011 11011 00011 00011 10010 10010 10010 10010 10010 10010 00011 00011
00011 00011 11011;

        3520 : 11011 11011 11011 11011 10010 10101 00101 00101 00101 00101 10101 10010 00011
00011 11011 11011;

        3504 : 11011 11011 11011 10010 10101 00101 00101 00101 00101 00101 00101 10101 10010
11011 11011 11011;

        3488 : 11011 11011 11011 10010 00101 00101 00101 00101 00101 00101 00101 00101 10010
11011 11011 11011;

        3472 : 11011 11011 11011 10010 10101 00101 00101 00101 00101 00101 00101 10101 10010
11011 11011 11011;

        3456 : 11011 11011 11011 10101 00101 10110 10110 10110 10110 10110 00011 00011 00011
11011 11011 11011;

        3440 : 11011 11011 10101 10110 00011 00011 00011 10101 10101 00011 00011 11011 11011
00011 11011 11011;

        3424 : 11011 00011 00101 00011 11011 11011 11011 00011 00011 00011 00011 11011 11011
00011 11011 11011;

        3408 : 00011 11010 00101 00011 11011 11011 11011 11011 00011 00011 00011 00011 00011
00011 11011 11011;

        3392 : 11010 11010 11010 00011 00011 11011 11011 00011 00011 00011 00011 00011 00011
11011 11011 11011;
```

```
        3376 : 11010 11011 11010 11010 00011 00011 00011 00011 00011 11011 11011 11011 00011
11011 11011 11011;

        3360 : 11011 11011 11010 11011 00011 00011 00011 00011 00011 11011 11011 11011 00011
11011 11011 11011;

        3344 : 11011 11011 11011 11011 00011 00011 00011 00011 00011 00011 11011 11011 00011
11011 11011 11011;

        3328 : 11011 11011 11011 11011 11011 00011 00011 00011 11011 00011 00011 00011 11011
11011 11011 11011;

        3312 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

        3296 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

        3280 : 01100 11100 11100 11100 11100 11100 11100 01100 00111 00111 01100 11100 11100
11100 11100 11100;

        3264 : 00111 00111 01100 11100 01100 00111 00111 00111 00011 00011 00111 00111 01100
11100 11100 11100;

        3248 : 00011 00011 00111 00111 00111 00011 00011 00011 00011 00011 00011 00011 00111
01100 11100 11100;

        3232 : 00011 00011 00011 00011 00011 00011 00011 00011 11011 11011 00011 00011 00011
00111 11100 11100;

        3216 : 11011 11010 00011 00011 00011 11011 11011 11011 11011 11011 11011 11011 00011
00111 11100 11100;

        3200 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011
00111 11100 11100;

        3184 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 00011
00111 11100 11100;

        3168 : 11011 00011 00011 00011 00011 11011 11011 11011 11011 11011 11011 11011 00111
01100 11100 11100;

        3152 : 00011 11010 00011 11010 11010 00011 11011 11011 11011 11011 11011 11011 00111
01100 11100 11100;

        3136 : 11010 11010 11010 11010 11010 11010 11011 11011 11011 11011 11011 11011 00011
00111 11100 11100;

        3120 : 11010 11011 11010 11010 11011 11010 11011 11011 11011 11011 11011 11011 00011
00111 11100 11100;
```

```
	3104 : 11011 11011 11010 11011 11011 11011 11011 11011 11011 11011 11011 11011 00111 01100 11100 11100;

	3088 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 00111 11100 11100 11100;

	3072 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 00111 11100 11100 11100;

	3056 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	3040 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	3024 : 11100 11100 11100 11100 11100 01100 00111 00111 00111 00111 01100 11100 01100 00111 00111 00111;

	3008 : 11100 11100 11100 01100 00111 00111 00011 00011 00011 00011 00111 00111 00111 00011 00011 00011;

	2992 : 11100 11100 01100 00111 00011 00011 00011 00011 00011 00011 00011 00011 00011 00011 00011 00011;

	2976 : 11100 11100 00111 00011 00011 00011 11011 00011 11011 11011 00011 00011 00011 11010 11011 11011;

	2960 : 11100 11100 00111 00011 11011 11010 11011 11010 11011 11011 11011 11011 11011 11011 11011 11011;

	2944 : 11100 11100 00111 11011 11011 11011 11011 11010 11011 11011 11011 11011 11011 11011 11011 11011;

	2928 : 11100 11100 00111 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	2912 : 11100 11100 01100 00111 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	2896 : 11100 11100 11100 01100 00111 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	2880 : 11100 11100 11100 01100 00111 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	2864 : 11100 11100 11100 00111 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	2848 : 11100 11100 11100 00111 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;
```

```
2832 : 11100 11100 01100 00111 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

2816 : 11100 11100 00111 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

2800 : 11011 00011 11011 11011 00011 00111 00011 11011 00011 11011 11011 00011 11011 00011 11011 11011;

2784 : 11011 11011 00011 00011 00011 00011 00011 00011 00011 00011 11011 11011 00011 11011 11011 11011;

2768 : 11011 11011 00011 11011 00011 00011 00011 00011 11011 00011 00011 00011 00011 11011 11011 11011;

2752 : 11011 11011 00011 00011 11010 00011 00011 11010 00011 11011 11010 11011 11010 11011 11011 11011;

2736 : 11011 00011 11011 11011 00011 11010 00011 00011 00011 00011 00011 11011 11010 00011 11011 11011;

2720 : 00011 11011 11011 00011 00011 00011 00011 00011 00011 00011 11011 00011 11011 11010 11011 11011;

2704 : 11011 00011 11011 11011 00011 00011 00011 00011 00011 11011 00011 11011 00011 11011 11011 11011;

2688 : 11011 11011 00011 00011 00110 11011 00011 00011 00011 00011 11011 11011 00011 11011 11011 11011;

2672 : 11011 11011 11011 00111 01100 00111 00011 00011 00011 11010 00110 00011 11010 00011 11011 11011;

2656 : 11011 00011 00011 00011 00111 11011 00011 00011 00011 00111 01100 00111 11010 11010 11011 11011;

2640 : 00011 11010 00011 11010 00011 00011 00011 00011 11011 00011 00111 11011 00011 11010 11011 11011;

2624 : 11010 11010 11010 11010 00011 11011 00011 00011 11011 00011 11011 00011 11011 00011 11011 11011;

2608 : 11010 11011 11010 11010 00011 11011 11011 00011 00011 11011 00011 11011 11011 00011 11011 11011;

2592 : 11011 11011 11010 11011 11011 00011 00011 11011 11011 11011 11011 00011 00011 11011 11011 11011;

2576 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;
```

```
2560 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

2544 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 00111 01100 11100 11100;

2528 : 11011 11011 11011 11011 11010 11011 11011 11011 11011 11011 11011 11011 00011 00111 11100 11100;

2512 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 00011 00111 11100 11100;

2496 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 00111 11100 11100;

2480 : 11011 11011 11011 11011 11011 11011 00011 11011 00011 11011 11011 11011 00011 00111 11100 11100;

2464 : 11011 11011 11011 11011 11011 11011 11010 11011 11010 11011 11011 11011 00111 01100 11100 11100;

2448 : 11011 11011 11011 11011 11011 11011 11011 11011 11010 00011 11011 11011 00111 01100 11100 11100;

2432 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11010 11011 11011 00011 00111 11100 11100;

2416 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 00011 00111 11100 11100;

2400 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 00111 11100 11100;

2384 : 11011 11011 11011 11011 11011 11011 11011 11011 11010 11011 11011 11011 00011 00111 11100 11100;

2368 : 11011 00011 00111 00111 00111 00011 11011 11011 11011 11011 11011 00011 00111 01100 11100 11100;

2352 : 00111 00111 01100 11100 01100 00111 00111 00111 00011 00011 00111 00111 01100 11100 11100 11100;

2336 : 01100 11100 11100 11100 11100 11100 11100 01100 00111 00111 01100 11100 11100 11100 11100 11100;

2320 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

2304 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;
```

```
        2288 : 11100 11100 00111 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        2272 : 11100 11100 01100 00111 00011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        2256 : 11100 11100 11100 01100 00111 11011 11011 11011 11011 11011 11011 00011 11011
11011 11011 11011;

        2240 : 11100 11100 11100 01100 00111 11011 11011 11011 11011 11011 11011 11010 11011
11011 11011 11011;

        2224 : 11100 11100 11100 00111 00011 11011 11011 11011 11011 11011 11011 11010 11011
11011 11011 11011;

        2208 : 11100 11100 11100 00111 00011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        2192 : 11100 11100 01100 00111 11011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        2176 : 11100 11100 00111 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        2160 : 11100 11100 00111 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011
11010 11011 11011;

        2144 : 11100 11100 00111 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        2128 : 11100 11100 00111 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        2112 : 11100 11100 11100 00111 00011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        2096 : 11100 11100 11100 11100 00111 00111 00011 11011 11011 00011 00111 00111 00111
00111 00011 00011;

        2080 : 11100 11100 11100 11100 11100 11100 00111 00111 00111 00111 01100 11100 11100
01100 00111 00111;

        2064 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

        2048 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

        2032 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;
```

```
2016 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

2000 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

1984 : 11011 11011 11011 11011 00011 11011 11011 00011 11011 00011 00011 11011 11011 11011 11011 11011;

1968 : 11011 11011 11011 00011 11011 11011 11011 11011 00011 11011 00011 00011 11011 11011 11011 11011;

1952 : 11011 11011 11011 00011 11011 00011 11011 00011 11011 11011 11011 00011 00011 11011 11011 11011;

1936 : 11011 11011 00011 11011 00011 00011 00011 00011 11011 11011 11011 00011 11011 00011 11011 11011;

1920 : 11011 11011 11011 11011 11011 00011 11011 11011 11011 11011 11011 11011 00011 11011 11011 11011;

1904 : 11011 11011 11011 00011 11011 11011 11011 00011 00011 11011 11011 11011 00011 11011 11011 11011;

1888 : 11011 11011 11011 11011 11011 11011 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

1872 : 11011 11011 11011 11011 11011 11011 11011 11011 00011 11011 00011 00011 11011 11011 11011 11011;

1856 : 11011 00011 11011 11011 11011 11011 11011 00011 11011 00011 00011 11011 11011 11011 11011 11011;

1840 : 11011 11010 11011 11011 11011 11011 11011 11011 11011 11011 11011 00011 11011 11011 11011 11011;

1824 : 11011 11010 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

1808 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

1792 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

1776 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

1760 : 11011 11011 11011 11011 11010 11011 00011 00011 00011 00011 00011 11011 11011 11011 11011 11011;
```

```
	1744 : 11011 11011 11011 00011 10010 10010 00011 10010 10010 10010 10010 00011 00011 00011 11011 11011;

	1728 : 11011 11011 11011 10010 11101 10010 10010 11101 11101 11101 10010 10010 00011 00011 00011 11011;

	1712 : 11011 11011 10010 11101 11101 11101 10110 10110 10010 10110 10110 11101 11101 00011 11011 11011;

	1696 : 11011 11011 10010 10010 11101 10010 10110 10010 10010 10010 00110 00110 10110 11101 11011 11011;

	1680 : 11011 11011 11011 10010 10110 10110 10110 10010 10010 00110 00110 00110 10110 11101 11011 11011;

	1664 : 11011 11011 11101 10110 10110 10010 10010 00110 00110 00110 00110 00110 10110 11101 11011 11011;

	1648 : 11011 11011 11101 10110 10010 10010 10010 00110 00110 00110 00110 00110 10110 11101 11011 11011;

	1632 : 11011 00011 11101 10110 00110 10010 00110 00110 00110 00110 00110 00110 11101 11011 11011 11011;

	1616 : 00011 11010 11010 11101 00110 00110 00110 00110 00110 00110 00110 10110 11011 11011 11011 11011;

	1600 : 11010 11010 11010 11010 10110 00110 00110 00110 00110 00110 10110 11011 00011 11011 11011 11011;

	1584 : 11010 11011 11010 11010 11011 10110 10110 10110 10110 10110 11011 11011 11010 11011 11011 11011;

	1568 : 11011 11011 11010 11011 11011 11011 11011 11011 11011 11011 11011 11011 11010 11011 11011 11011;

	1552 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	1536 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	1520 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	1504 : 11011 11011 11011 11011 00011 00011 00011 00011 11011 11011 11011 11011 11011 11011 11011 11011;

	1488 : 11011 11011 11011 00011 11010 00011 11010 11010 00011 11011 11011 00011 11011 11011 11011 11011;
	1472 : 11011 11011 11011 11010 11010 11010 11010 11010 11010 11011 11011 11010 11011 11011 11011 11011;

	1456 : 11011 11011 11011 11010 11011 11010 11010 11011 11010 11011 11011 11010 11011 11011 11011 11011;

	1440 : 11011 11011 11011 11011 11011 11010 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	1424 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	1408 : 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	1392 : 11010 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11010 11011 11011;

	1376 : 11010 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	1360 : 11011 11010 11011 11011 00011 11011 00011 00111 00111 00011 11011 11011 11011 11011 11011 11011;

	1344 : 11011 11011 00011 00111 00111 00111 00111 01100 01100 00111 00111 11011 11011 11011 11011 00011;

	1328 : 00111 00111 00111 01100 11100 11100 11100 11100 11100 11100 11100 00111 00111 00111 00111 00111;

	1312 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	1296 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	1280 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	1264 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	1248 : 11011 11011 11011 11011 00011 00011 00011 00011 11011 11011 11011 11011 11011 11011 11011 11011;

	1232 : 11011 11011 11011 00011 11010 00011 11010 11010 00011 11011 11011 00011 11011 11011 11011 11011;

	1216 : 11011 11011 11011 11010 11010 11010 11010 11010 11010 11011 11011 11010 11011 11011 11011 11011;
```

```
        1200 : 11011 11011 11011 11010 11011 11010 11010 11011 11010 11011 11011 11010 11011
11011 11011 11011;

        1184 : 11011 11011 11011 11011 11011 11010 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        1168 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        1152 : 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        1136 : 11010 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011
11010 11011 11011;

        1120 : 11010 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        1104 : 11011 11010 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        1088 : 11011 11011 11011 11011 11011 11011 11011 00011 11011 11010 11011 11011 11011
11011 11011 11011;

        1072 : 11011 11011 11011 11011 11011 11011 11011 11010 11011 11010 11011 11011 11011
11011 11011 11011;

        1056 : 11011 11011 11011 11011 11011 11011 00011 11010 11011 11011 11011 11011 11011
11011 11011 11011;

        1040 : 11011 11011 11011 11011 11011 11011 11010 11011 11011 11011 11011 11010 11011
11011 11011 11011;

        1024 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        1008 : 11100 11100 11100 00111 00011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        992 : 11100 11100 01100 00111 11011 11011 00011 00011 11011 11011 11011 11011 11011
11011 11011 11011;

        976 : 11100 11100 00111 00011 11011 11011 11010 11010 00011 11011 11011 11011 00011
11011 11011 11011;

        960 : 11100 11100 00111 00011 11011 11011 11011 11010 11010 11011 11010 11011 11010
11011 11011 11011;

        944 : 11100 11100 00111 11011 11011 11011 11011 11011 11010 11011 11011 11011 11010
00011 11011 11011;
```

```
	928 : 11100 11100 00111 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11010 11011 11011;

	912 : 11100 11100 00111 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	896 : 11100 11100 01100 00111 11011 11011 11011 11011 11011 00011 00011 00011 00011 11011 11011 11011;

	880 : 11100 11100 01100 00111 11011 11011 11011 11011 00011 11010 11010 00011 11010 00011 11011 11011;

	864 : 11100 11100 00111 00011 11011 11011 11011 11011 11010 11010 11010 11010 11010 11010 11011 11011;

	848 : 11100 11100 00111 00011 11010 00011 11011 11011 11010 11011 11010 11010 11011 11010 11011 11011;

	832 : 11100 11100 00111 11011 11010 11010 11011 11011 11011 11011 11010 11011 11011 11011 11011 11011;

	816 : 11100 11100 00111 11011 11011 11010 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	800 : 11100 11100 00111 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	784 : 11100 11100 11100 00111 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	768 : 11100 11100 11100 00111 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	752 : 11011 11011 11011 11011 11011 11011 11011 11011 00011 11011 00011 00011 00111 11100 11100 11100;

	736 : 11011 11011 11011 11011 11011 11011 11011 11011 11010 11011 11010 11011 00111 11100 11100 11100;

	720 : 11011 11011 11011 11011 11011 11011 11011 00011 11010 11011 11010 11011 00111 11100 11100 11100;

	704 : 11011 11011 11011 11011 11011 11011 11011 11010 11011 11011 11010 11011 00111 11100 11100 11100;

	688 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 00111 11100 11100 11100;

	672 : 11011 11010 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 00111 11100 11100 11100;
```

```
        656 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 00011 00111
11100 11100 11100;

        640 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 00111 01100
11100 11100 11100;

        624 : 11011 11011 11011 00011 11011 00011 11011 11011 11011 11011 00111 01100 11100
11100 11100 11100;

        608 : 11011 11011 11011 11010 11011 11010 11011 11011 11011 11011 00111 11100 11100
11100 11100 11100;

        592 : 11011 11011 11011 11010 11011 11010 00011 11011 11011 11011 00111 11100 11100
11100 11100 11100;

        576 : 11011 11011 11011 11010 11011 11011 11010 11011 11011 11011 00111 01100 11100
11100 11100 11100;

        560 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 00011 00111 11100
11100 11100 11100;

        544 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 00011 00011 00111
11100 11100 11100;

        528 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 00011 00111
11100 11100 11100;

        512 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 00011 00111
11100 11100 11100;

        496 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        480 : 11011 11011 11011 11011 11010 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        464 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011
11011 11011 11011;

        448 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11010 11011
11011 11011 11011;

        432 : 11011 11011 11011 11011 11011 11011 00011 11011 00011 11011 11011 11011 11011
11011 11011 11011;

        416 : 11011 11011 11011 11011 11011 11011 11010 11011 11010 11011 11011 11011 11011
11011 11011 11011;

        400 : 11011 11011 11011 11011 11011 11011 11011 11011 11010 00011 11011 11011 11011
11011 11011 11011;
```

```
	384 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11010 11011 11011 11011 11011 11011 11011;

	368 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	352 : 11011 00011 00011 00011 00011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	336 : 00011 11010 11010 00011 11010 00011 11011 11011 11010 11011 11011 11011 11011 11011 11011 11011;

	320 : 11010 11010 11010 11010 11010 11010 11011 11011 11011 11011 11011 11011 00011 11011 11011 11011;

	304 : 11010 11011 11010 11010 11011 11010 11011 11011 11011 11011 11011 11011 11010 11011 11011 11011;

	288 : 11011 11011 11010 11011 11011 11011 11011 11011 11011 11011 11011 11011 11010 11011 11011 11011;

	272 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	256 : 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011 11011;

	240 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	224 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	208 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	192 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	176 : 11100 11100 11100 11100 11100 01100 11100 11100 11100 11100 01100 11100 11100 11100 11100 11100;

	160 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100;

	144 : 11100 11100 01100 01100 11100 11100 11100 11100 11100 11100 11100 11100 01100 01100 11100 11100;

	128 : 11100 01100 01100 01100 01100 11100 11100 11100 11100 11100 01100 01100 01100 01100 01100 11100;
	112 : 11100 01100 01100 00111 00111 01100 01100 01100 01100 01100 01100 00111 00111
01100 01100 11100;

	96 : 11100 11100 11100 11100 01100 00111 00111 01100 01100 00111 00111 01100 11100
11100 11100 11100;

	80 : 11100 11100 11100 11100 11100 11100 01100 00111 00111 01100 11100 11100 11100
11100 11100 11100;

	64 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

	48 : 11100 11100 11100 11100 11100 01100 11100 11100 11100 11100 01100 11100 11100
11100 11100 11100;

	32 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

	16 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

	0 : 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100 11100
11100 11100 11100;

END;
```

tileROM.v

```
// megafunction wizard: %ROM: 1-PORT%

// GENERATION: STANDARD

// VERSION: WM1.0

// MODULE: altsyncram


// ============================================================

// File Name: tileROM.v

// Megafunction Name(s):

// 			altsyncram

//
```

```
// Simulation Library Files(s):
//                              altera_mf
// ============================================================
// ************************************************************
// THIS IS A WIZARD-GENERATED FILE. DO NOT EDIT THIS FILE!
//
// 21.1.0 Build 842 10/21/2021 SJ Lite Edition
// ************************************************************


//Copyright (C) 2021  Intel Corporation. All rights reserved.
//Your use of Intel Corporation's design tools, logic functions
//and other software and tools, and any partner logic
//functions, and any output files from any of the foregoing
//(including device programming or simulation files), and any
//associated documentation or information are expressly subject
//to the terms and conditions of the Intel Program License
//Subscription Agreement, the Intel Quartus Prime License Agreement,
//the Intel FPGA IP License Agreement, or other applicable license
//agreement, including, without limitation, that your use is for
//the sole purpose of programming logic devices manufactured by
//Intel and sold by Intel or its authorized distributors.  Please
//refer to the applicable agreement for further details, at
//https://fpgasoftware.intel.com/eula.


// synopsys translate_off
`timescale 1 ps / 1 ps
// synopsys translate_on
```

```
module tileROM (
	address,
	clock,
	q);

	input	[12:0]  address;
	input	  clock;
	output	[4:0]  q;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_off
`endif
	tri1	  clock;
`ifndef ALTERA_RESERVED_QIS
// synopsys translate_on
`endif

	wire [4:0] sub_wire0;
	wire [4:0] q = sub_wire0[4:0];

	altsyncram	altsyncram_component (
				.address_a (address),
				.clock0 (clock),
				.q_a (sub_wire0),
				.aclr0 (1'b0),
				.aclr1 (1'b0),
				.address_b (1'b1),
				.addressstall_a (1'b0),
				.addressstall_b (1'b0),
				.byteena_a (1'b1),
                                        .byteena_b (1'b1),
                                        .clock1 (1'b1),
                                        .clocken0 (1'b1),
                                        .clocken1 (1'b1),
                                        .clocken2 (1'b1),
                                        .clocken3 (1'b1),
                                        .data_a ({5{1'b1}}),
                                        .data_b (1'b1),
                                        .eccstatus (),
                                        .q_b (),
                                        .rden_a (1'b1),
                                        .rden_b (1'b1),
                                        .wren_a (1'b0),
                                        .wren_b (1'b0));
        defparam
                altsyncram_component.address_aclr_a = "NONE",
                altsyncram_component.clock_enable_input_a = "BYPASS",
                altsyncram_component.clock_enable_output_a = "BYPASS",
                altsyncram_component.init_file = "mif/background.mif",
                altsyncram_component.intended_device_family = "Cyclone V",
                altsyncram_component.lpm_hint = "ENABLE_RUNTIME_MOD=NO",
                altsyncram_component.lpm_type = "altsyncram",
                altsyncram_component.numwords_a = 8192,
                altsyncram_component.operation_mode = "ROM",
                altsyncram_component.outdata_aclr_a = "NONE",
                altsyncram_component.outdata_reg_a = "UNREGISTERED",
                altsyncram_component.ram_block_type = "M10K",
                altsyncram_component.widthad_a = 13,
                altsyncram_component.width_a = 5,
		altsyncram_component.width_byteena_a = 1;



endmodule


// ============================================================
// CNX file retrieval info
// ============================================================
// Retrieval info: PRIVATE: ADDRESSSTALL_A NUMERIC "0"
// Retrieval info: PRIVATE: AclrAddr NUMERIC "0"
// Retrieval info: PRIVATE: AclrByte NUMERIC "0"
// Retrieval info: PRIVATE: AclrOutput NUMERIC "0"
// Retrieval info: PRIVATE: BYTE_ENABLE NUMERIC "0"
// Retrieval info: PRIVATE: BYTE_SIZE NUMERIC "8"
// Retrieval info: PRIVATE: BlankMemory NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_INPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: CLOCK_ENABLE_OUTPUT_A NUMERIC "0"
// Retrieval info: PRIVATE: Clken NUMERIC "0"
// Retrieval info: PRIVATE: IMPLEMENT_IN_LES NUMERIC "0"
// Retrieval info: PRIVATE: INIT_FILE_LAYOUT STRING "PORT_A"
// Retrieval info: PRIVATE: INIT_TO_SIM_X NUMERIC "0"
// Retrieval info: PRIVATE: INTENDED_DEVICE_FAMILY STRING "Cyclone V"
// Retrieval info: PRIVATE: JTAG_ENABLED NUMERIC "0"
// Retrieval info: PRIVATE: JTAG_ID STRING "NONE"
// Retrieval info: PRIVATE: MAXIMUM_DEPTH NUMERIC "0"
// Retrieval info: PRIVATE: MIFfilename STRING "./river_raid_hw/mif/background.mif"
// Retrieval info: PRIVATE: NUMWORDS_A NUMERIC "8192"
// Retrieval info: PRIVATE: RAM_BLOCK_TYPE NUMERIC "2"
// Retrieval info: PRIVATE: RegAddr NUMERIC "1"
// Retrieval info: PRIVATE: RegOutput NUMERIC "0"
// Retrieval info: PRIVATE: SYNTH_WRAPPER_GEN_POSTFIX STRING "0"
// Retrieval info: PRIVATE: SingleClock NUMERIC "1"
// Retrieval info: PRIVATE: UseDQRAM NUMERIC "0"
// Retrieval info: PRIVATE: WidthAddr NUMERIC "13"
// Retrieval info: PRIVATE: WidthData NUMERIC "5"
// Retrieval info: PRIVATE: rden NUMERIC "0"
// Retrieval info: LIBRARY: altera_mf altera_mf.altera_mf_components.all
// Retrieval info: CONSTANT: ADDRESS_ACLR_A STRING "NONE"
// Retrieval info: CONSTANT: CLOCK_ENABLE_INPUT_A STRING "BYPASS"
// Retrieval info: CONSTANT: CLOCK_ENABLE_OUTPUT_A STRING "BYPASS"
// Retrieval info: CONSTANT: INIT_FILE STRING "./river_raid_hw/mif/background.mif"
// Retrieval info: CONSTANT: INTENDED_DEVICE_FAMILY STRING "Cyclone V"
// Retrieval info: CONSTANT: LPM_HINT STRING "ENABLE_RUNTIME_MOD=NO"
// Retrieval info: CONSTANT: LPM_TYPE STRING "altsyncram"
// Retrieval info: CONSTANT: NUMWORDS_A NUMERIC "8192"
// Retrieval info: CONSTANT: OPERATION_MODE STRING "ROM"
// Retrieval info: CONSTANT: OUTDATA_ACLR_A STRING "NONE"
// Retrieval info: CONSTANT: OUTDATA_REG_A STRING "UNREGISTERED"
// Retrieval info: CONSTANT: RAM_BLOCK_TYPE STRING "M10K"
// Retrieval info: CONSTANT: WIDTHAD_A NUMERIC "13"
// Retrieval info: CONSTANT: WIDTH_A NUMERIC "5"
// Retrieval info: CONSTANT: WIDTH_BYTEENA_A NUMERIC "1"
// Retrieval info: USED_PORT: address 0 0 13 0 INPUT NODEFVAL "address[12..0]"
// Retrieval info: USED_PORT: clock 0 0 0 0 INPUT VCC "clock"
// Retrieval info: USED_PORT: q 0 0 5 0 OUTPUT NODEFVAL "q[4..0]"
// Retrieval info: CONNECT: @address_a 0 0 13 0 address 0 0 13 0
// Retrieval info: CONNECT: @clock0 0 0 0 0 clock 0 0 0 0
// Retrieval info: CONNECT: q 0 0 5 0 @q_a 0 0 5 0
// Retrieval info: GEN_FILE: TYPE_NORMAL tileROM.v TRUE

// Retrieval info: GEN_FILE: TYPE_NORMAL tileROM.inc FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL tileROM.cmp FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL tileROM.bsf FALSE

// Retrieval info: GEN_FILE: TYPE_NORMAL tileROM_inst.v TRUE

// Retrieval info: GEN_FILE: TYPE_NORMAL tileROM_bb.v TRUE

// Retrieval info: LIB_FILE: altera_mf
```