ASIC | 专注开源硬件 IP Core
www.openasic.org

帧内预测

作者 陈珂

日期 2017.1.13

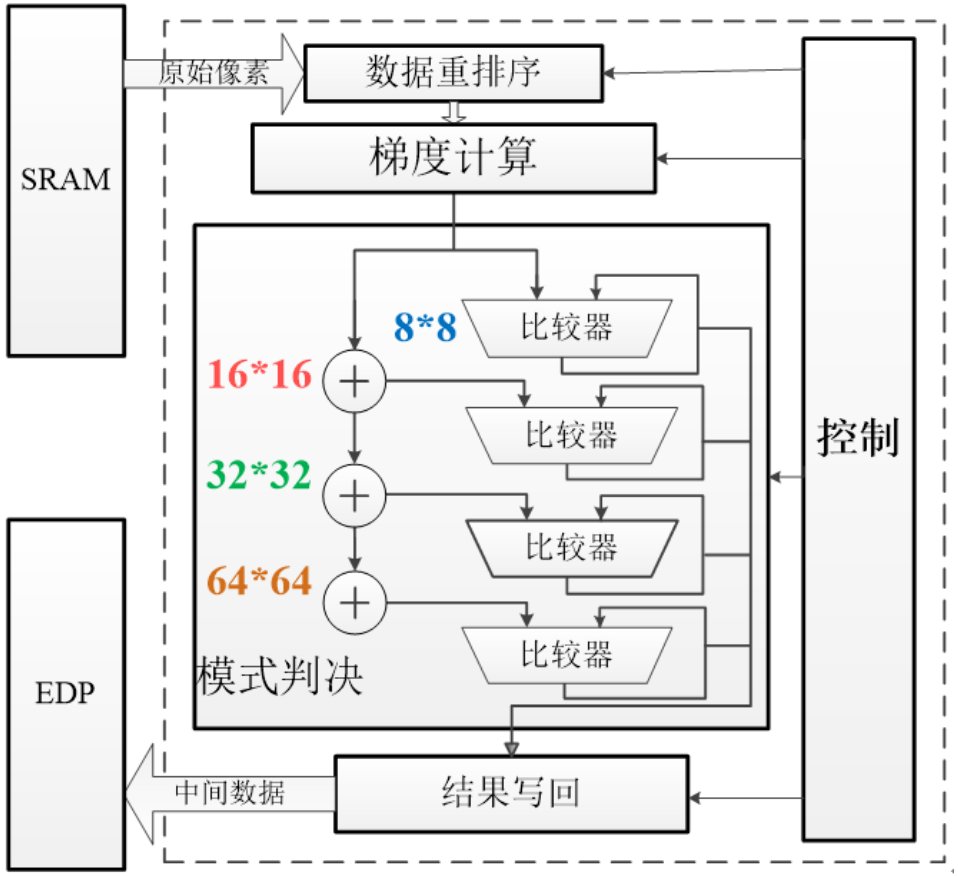
帧内预测

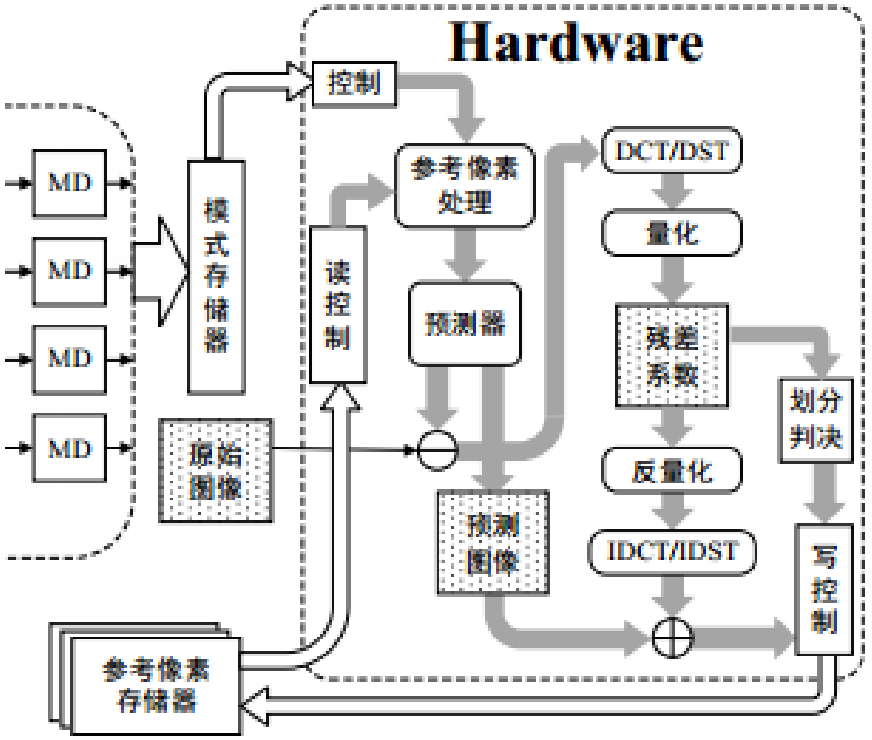
- 硬件架构
- 硬件接口
- Testbench详解
- 仿真步骤
- 学习资料

帧内预测

- 硬件架构
- 硬件接口
- Testbench详解
- 仿真步骤
- 学习资料

硬件架构

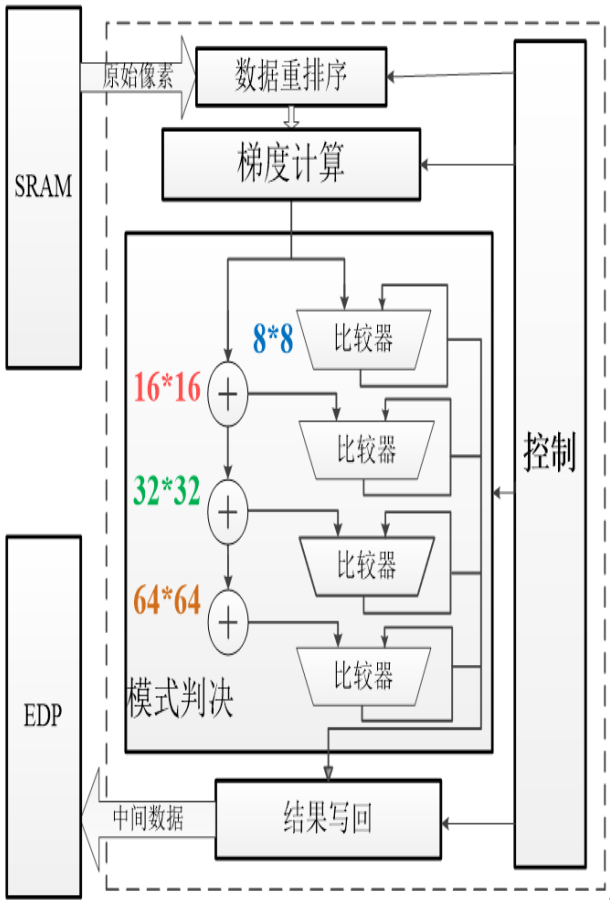


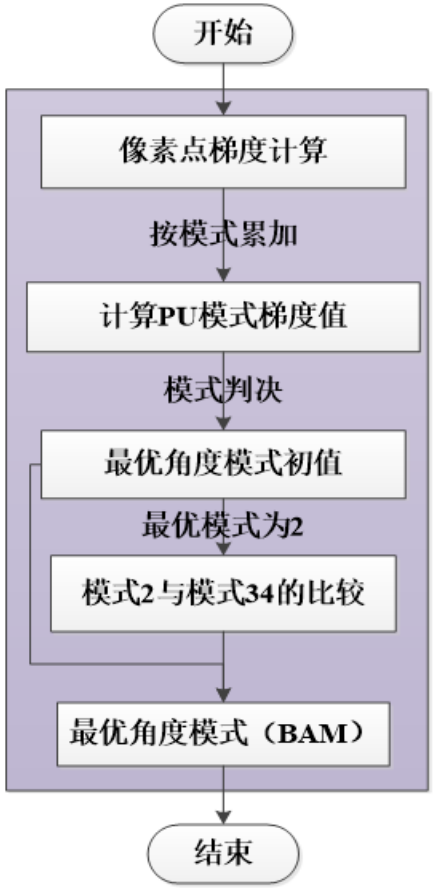


Pre_intra：硬件加速单元

Intra：帧内预测过程

硬件架构——pre_intra

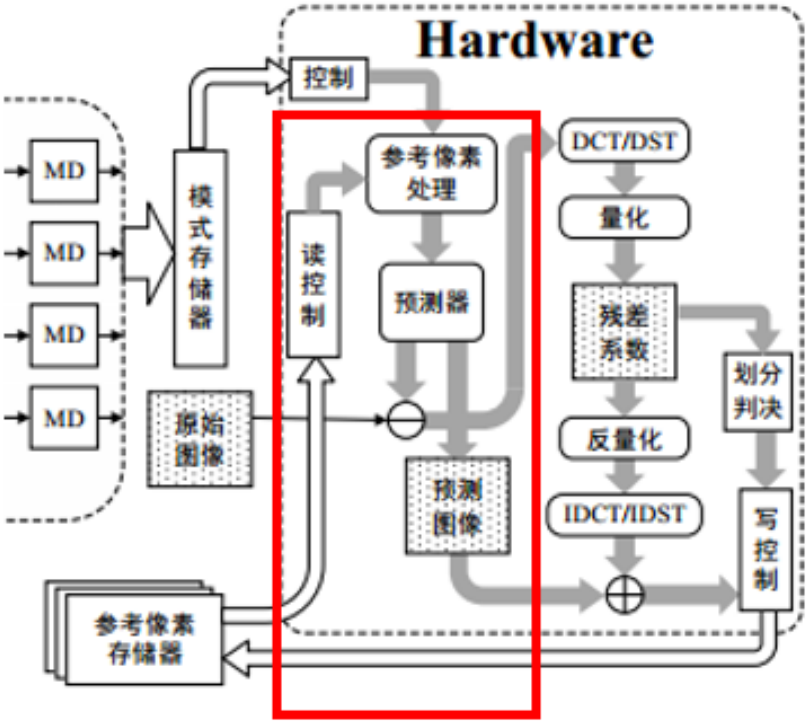




Pre_intra：模式判决

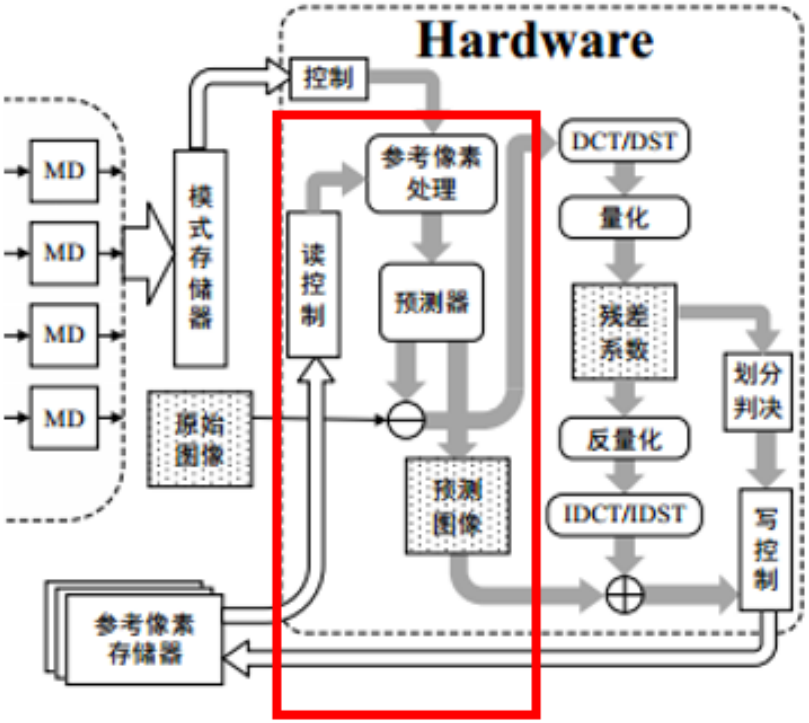
- 数据重排序
- 梯度计算
- 模式判决
- 结果写回
- 模块控制

硬件架构——intra



- 参考像素处理：填充、滤波等
- 预测：计算残差值的预测像素
- 模式判断：35种预测模式
- 块划分判断：最优划分

硬件架构——intra



- 参考像素处理：填充、滤波等
- 预测：计算残差值的预测像素
- 模式判断：35种预测模式
- 块划分判断：最优划分

帧内预测

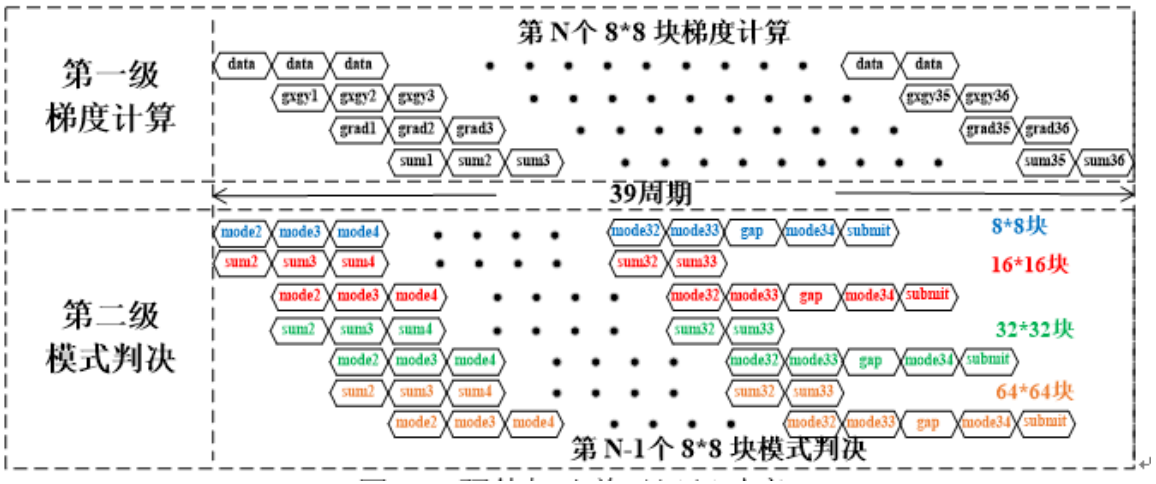
- 硬件架构
- 硬件接口
- Testbench详解
- 仿真步骤
- 学习资料

硬件接口——pre_intra

Hevc_md_top

- md_top
  1. md_fetch：data fetch
  2. control: module controller
  3. gxgy: gx、gy calculation
  4. counter: gradient value calculation
  5. compare: best angle mode decision
  6. dc_planar: mode decision(planar/DC/best angular mode)
- md_ram: original pixels data ram
- Fetch8x8: 8x8 original pixels
- mode_write: mode write back to mode ram

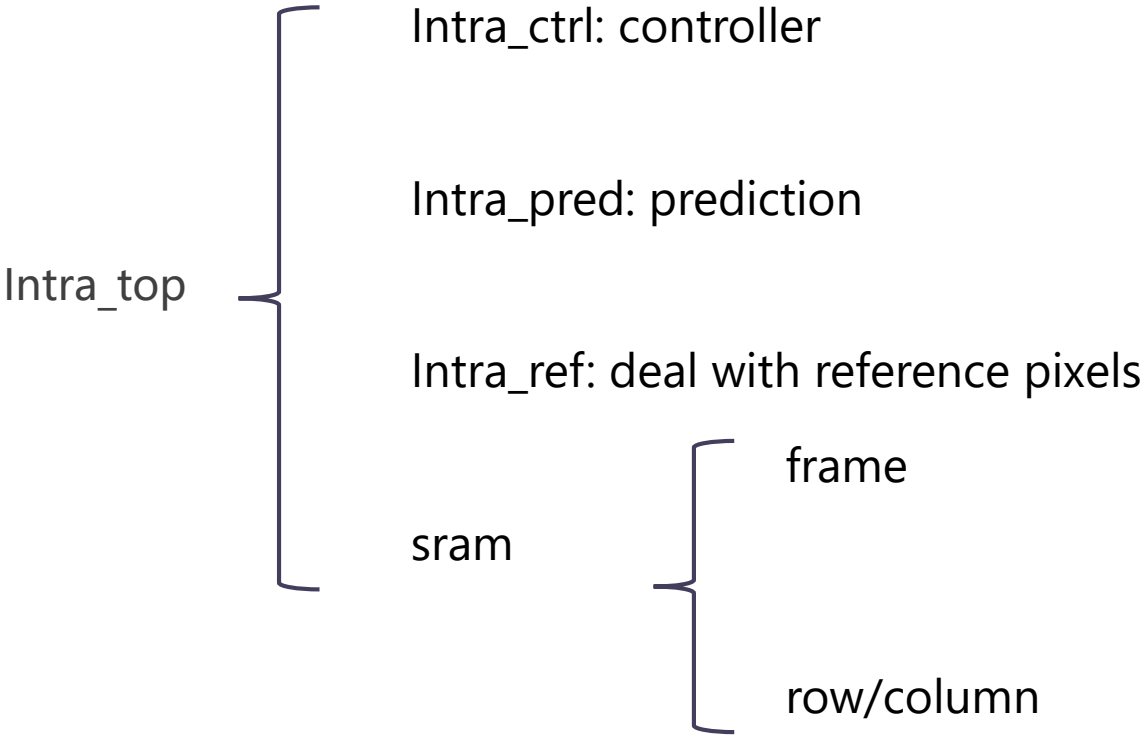
硬件接口——pre_intra



1. md_fetch：data fetch
2. control: module controller
3. gxgy: gx、gy calculation
4. counter: gradient value calculation
5. compare: best angle mode decision
6. dc_planar: mode decision(planar/DC/best angular mode)

- 两级流水线结构
- 数据重排序&梯度计算
  - 数据读取
  - 每个像素点的梯度值计算
  - 每个像素点的模式梯度值计算
  - 模式梯度叠加
- 模式判决&结果写回

硬件接口——intra



硬件接口


```
module hevc_md_top(

    clk,
    rstn,

    md_ren_o,
    md_sel_o,
    md_size_o,
    md_4x4_x_o,
    md_4x4_y_o,
    md_idx_o,
    md_data_i,

    md_we,
    md_waddr,
    md_wdata,

    enable,
    finish

);
```

- 给出mode相关信号，读入初始像素值
- 将预测模式写入ram中
- 控制信号

```
module intra_top(
  clk                ,
  rst_n              ,
  // ctrl if
  pre_min_size_i     ,
  uv_partition_i     ,
  mb_x_total_i       ,
  mb_x_i             ,
  mb_y_i             ,
  start_i            ,
  done_o             ,
  // pre mode if
  md_rden_o          ,
  md_raddr_o         ,
  md_rdata_i         ,
  // tq pred if
  pre_en_o           ,
  pre_sel_o          ,
  pre_size_o         ,
  pre_4x4_x_o        ,
  pre_4x4_y_o        ,
  pre_data_o         ,
  pre_mode_o         ,
  pre_islast_o       ,
  pre_isfrst_o       ,
  pre_num_o      ,
// tq rec if
  rec_val_i          ,
  rec_idx_i          ,
  rec_data_i         ,
  // pt if
  skip_i
 );
```

- Ctrl if
  - 输入信号：最小是否为4x4的开关信号、分割信号、当前模块位置、开始信号
  - 输出信号：done
- Pre mode if：
  - 获取当前mode
- Tq pred if：
  - 输出pred相关结果
- Rec if：
  - 重建
  - Valid、index和重建后的像素值

帧内预测

- 硬件架构
- 硬件接口
- Testbench详解
- 仿真步骤
- 学习资料

testbench详解


```
intra_top u_intra_top(
  .clk               ( clk            ),
  .rst_n             ( rst_n          ),
  // ctrl if
  .pre_min_size_i    ( pre_min_size   ),
  .uv_partition_i    ( uv_partition   ),  // can be treated like a switch for chroma
  .mb_x_total_i      ( mb_x_total     ),
  .mb_x_i            ( mb_x           ),
  .mb_y_i            ( mb_y           ),
  .start_i           ( intra_start    ),
  .done_o            ( intra_done     ),
  // pre mode if
  .md_rden_o         ( md_rden        ),
  .md_raddr_o        ( md_raddr       ),
  .md_rdata_i        ( md_rdata       ),
  // tq pred if
  .pre_en_o          ( pre_en         ),
  .pre_sel_o         ( pre_sel        ),
  .pre_size_o        ( pre_size       ),
  .pre_4x4_x_o       ( pre_4x4_x      ),
  .pre_4x4_y_o       ( pre_4x4_y      ),
  .pre_data_o        ( pre_data       ),
  .pre_mode_o        ( pre_mode       ),
  // tq rec if                        //transition & qunant
  .rec_val_i         ( rec_val        ),
  .rec_idx_i         ( rec_idx        ),
  .rec_data_i        ( rec_data       ),
  .skip_i            ( skip_w         )
  );
```

- DUT : Intra_top
  - Ctrl If
  - Pre mode if
  - Tq pred if
  - Tq rec if

testbench详解


```
parameter DUMP_FILE          =    "intra_mode_cycopt.fsdb"          ;

parameter INPUT_MODE         =    "./tv/intra_mode.dat"             ;
parameter INPUT_MODE_UV      =    "./tv/intra_mode_uv.dat"          ;
parameter INPUT_MB_POSITION  =    "./tv/intra_mb_position.dat"      ;
parameter INPUT_MD_DECISION  =    "./tv/intra_md_decision.dat"      ;
parameter INPUT_4X4          =    "./tv/intra_rec4x4.dat"           ;
parameter INPUT_8X8          =    "./tv/intra_rec8x8.dat"           ;
parameter INPUT_16X16        =    "./tv/intra_rec16x16.dat"         ;
parameter INPUT_32X32        =    "./tv/intra_rec32x32.dat"         ;
parameter INPUT_4X4_U        =    "./tv/intra_rec4x4_u.dat"         ;
parameter INPUT_8X8_U        =    "./tv/intra_rec8x8_u.dat"         ;
parameter INPUT_16X16_U      =    "./tv/intra_rec16x16_u.dat"       ;
parameter INPUT_32X32_U      =    "./tv/intra_rec32x32_u.dat"       ;
parameter INPUT_4X4_V        =    "./tv/intra_rec4x4_v.dat"         ;
parameter INPUT_8X8_V        =    "./tv/intra_rec8x8_v.dat"         ;
parameter INPUT_16X16_V      =    "./tv/intra_rec16x16_v.dat"       ;
parameter INPUT_32X32_V      =    "./tv/intra_rec32x32_v.dat"       ;

parameter OUTPUT_FILE        =    "./dump/intra_pred_output.log"    ;

parameter CHECK_4X4          =    "./tv/intra_pre4x4.dat"           ;
parameter CHECK_8X8          =    "./tv/intra_pre8x8.dat"           ;
parameter CHECK_16X16        =    "./tv/intra_pre16x16.dat"         ;
parameter CHECK_32X32        =    "./tv/intra_pre32x32.dat"         ;
parameter CHECK_4X4_U        =    "./tv/intra_pre4x4_u.dat"         ;
parameter CHECK_8X8_U        =    "./tv/intra_pre8x8_u.dat"         ;
parameter CHECK_16X16_U      =    "./tv/intra_pre16x16_u.dat"       ;
parameter CHECK_32X32_U      =    "./tv/intra_pre32x32_u.dat"       ;
parameter CHECK_4X4_V        =    "./tv/intra_pre4x4_v.dat"         ;
parameter CHECK_8X8_V        =    "./tv/intra_pre8x8_v.dat"         ;
parameter CHECK_16X16_V      =    "./tv/intra_pre16x16_v.dat"       ;
parameter CHECK_32X32_V      =    "./tv/intra_pre32x32_v.dat"       ;
parameter INPUT_PARTITION_UV =    "./tv/intra_uv_partition.dat"     ;
parameter  MD_AMOUNT = 'd6 ;
```

- 输入文件test vector：
  - Input
  - Check
  - F265生成

testbench详解


```
task rec4x4 ;
  reg [`PIXEL_WIDTH*16-1:0] rec_data_16_pixel; //16pixel*8bit/pixel=128bit
  begin
    #10 ;
    rec_val    = 'd0 ;
    rec_idx    = 'd0 ;
    skip_w     = 'd0 ;
    rec_data   = 'd0 ;
    rec_val_en = 'd1 ;

  if(skip_w)
     rec_val_en = 'd0;
  else
     rec_val_en = 'd1;

  #10 skip_w    = 'd0 ;

  if(rec_val_en) begin
      #10 ;
        rec_val   = 'd1 ;
        rec_idx   = 'd0 ;

        if( u_intra_top.pre_sel_o==2'b00 ) begin//// 0: luma, 2: cb; 3:cr
          intra_tp  = $fscanf( fp_4x4_rec ,"%h" ,rec_data_16_pixel );
        end
        else if( u_intra_top.pre_sel_o==2'b10 )begin//2:cb
          intra_tp  = $fscanf( fp_8x8_rec_u ,"%h" ,rec_data_16_pixel );
        end
        else begin//3:cr
          intra_tp  = $fscanf( fp_8x8_rec_v ,"%h" ,rec_data_16_pixel );
        end

        rec_data  = ( rec_data_16_pixel<<(16*`PIXEL_WIDTH) );
    end

    #10 ;
    rec_val   = 'd0 ;
    rec_idx   = 'd0 ;
    skip_w = 'd0 ;
    rec_data  = 'd0 ;
  end
endtask
```

- Tq_rec_if
  - rec4x4
    - 4x4像素块的重建过程
    - 输入test vector中的重建像素值
  - rec8x8
  - rec16x16
  - rec 32x32

```
.rec_val_i          ( rec_val          ),
.rec_idx_i          ( rec_idx          ),
.rec_data_i         ( rec_data         ),
```

testbench详解


```
//--- Dump INTRA outputs ---------------

`ifdef DUMP_OUTPUT

  integer fp_out;

  initial begin
    fp_out = $fopen( OUTPUT_FILE ,"wb" );
  end

  always @(posedge clk) begin
    if( u_intra_top.pre_en_o ) begin
      $fdisplay(fp_out ,"====== Frame:%3d LCU x:%3d y:%3d position:%3d ======" ,
                frame_num ,mb_x ,mb_y ,u_intra_top.u_intra_ctrl.ref_position_o );

      $fdisplay(fp_out ,"%3d %3d %3d %3d",
                u_intra_top.u_intra_pred.pred_00_o ,u_intra_top.u_intra_pred.pred_01_o ,
                u_intra_top.u_intra_pred.pred_02_o ,u_intra_top.u_intra_pred.pred_03_o );

      $fdisplay(fp_out ,"%3d %3d %3d %3d",
                u_intra_top.u_intra_pred.pred_10_o ,u_intra_top.u_intra_pred.pred_11_o ,
                u_intra_top.u_intra_pred.pred_12_o ,u_intra_top.u_intra_pred.pred_13_o );

      $fdisplay(fp_out ,"%3d %3d %3d %3d",
                u_intra_top.u_intra_pred.pred_20_o ,u_intra_top.u_intra_pred.pred_21_o ,
                u_intra_top.u_intra_pred.pred_22_o ,u_intra_top.u_intra_pred.pred_23_o );

      $fdisplay(fp_out ,"%3d %3d %3d %3d",
                u_intra_top.u_intra_pred.pred_30_o ,u_intra_top.u_intra_pred.pred_31_o ,
                u_intra_top.u_intra_pred.pred_32_o ,u_intra_top.u_intra_pred.pred_33_o );
    end
  end

`endif
```

Dump出预测像素值

pred_00_o-pred_33_o

testbench详解


```
fp_check_4x4      = $fopen( CHECK_4X4      ,"r" );
fp_check_8x8      = $fopen( CHECK_8X8      ,"r" );
fp_check_16x16    = $fopen( CHECK_16X16    ,"r" );
fp_check_32x32    = $fopen( CHECK_32X32    ,"r" );
fp_check_4x4_u    = $fopen( CHECK_4X4_U    ,"r" );
fp_check_8x8_u    = $fopen( CHECK_8X8_U    ,"r" );
fp_check_16x16_u  = $fopen( CHECK_16X16_U  ,"r" );
fp_check_32x32_u  = $fopen( CHECK_32X32_U  ,"r" );
fp_check_4x4_v    = $fopen( CHECK_4X4_V    ,"r" );
fp_check_8x8_v    = $fopen( CHECK_8X8_V    ,"r" );
fp_check_16x16_v  = $fopen( CHECK_16X16_V  ,"r" );
fp_check_32x32_v  = $fopen( CHECK_32X32_V  ,"r" );
check_switch      = 3'b111 ; // 3'byuv
```

- Autocheck
- 输入check对照文件
- 与预测值对比

```
if( ((u_intra_top.pre_sel_o==2'b00)&&(check_switch[2]==1'b1)) ||//y
    ((u_intra_top.pre_sel_o==2'b10)&&(check_switch[1]==1'b1)) ||//u
    ((u_intra_top.pre_sel_o==2'b11)&&(check_switch[0]==1'b1))//v
  ) begin
  if( check_data !== { u_intra_top.u_intra_pred.pred_00_o ,u_intra_top.u_intra_pred.pred_01_o ,
                       u_intra_top.u_intra_pred.pred_02_o ,u_intra_top.u_intra_pred.pred_03_o ,
                       u_intra_top.u_intra_pred.pred_10_o ,u_intra_top.u_intra_pred.pred_11_o ,
                       u_intra_top.u_intra_pred.pred_12_o ,u_intra_top.u_intra_pred.pred_13_o ,
                       u_intra_top.u_intra_pred.pred_20_o ,u_intra_top.u_intra_pred.pred_21_o ,
                       u_intra_top.u_intra_pred.pred_22_o ,u_intra_top.u_intra_pred.pred_23_o ,
                       u_intra_top.u_intra_pred.pred_30_o ,u_intra_top.u_intra_pred.pred_31_o ,
                       u_intra_top.u_intra_pred.pred_32_o ,u_intra_top.u_intra_pred.pred_33_o }
    ) begin
    $display( "\n\t\t\t\t%8d, V_ERROR:pusize:%d at pux:%d puy:%d,it should be %x however pred_00_o is %x\n" ,$time
                                          ,pre_size,pre_4x4_x,pre_4x4_y,check_data,u_intra_top.u_intra_pred.pred_00_o
    #1000 ;
```

帧内预测

- 硬件架构
- 硬件接口
- Testbench详解
- 仿真步骤
- 学习资料

仿真步骤


```
File  Edit  View  Terminal  Tabs  Help
[chenke@vipdev2 sim]$ make ncsim
ncverilog +access+r -l ./dump/tb_intra.log -f file.f -timescale 1ns/1ps
ncverilog: 09.20-p007: (c) Copyright 1995-2009 Cadence Design Systems, Inc.
ncverilog: *W,LOGOPN: log file (./dump/tb_intra.log) could not be opened, contin
uing without it (No such file or directory).
include ../../../../../cds.lib
|
ncverilog: *W,DLCPTH (./INCA_libs/irun.lnx86.09.20.nc/cdsrun.lib,1): cds.lib No
such file '/home/chenke/cds.lib' (cds.lib command ignored).
include ../../../../../cds.lib
|
ncverilog: *W,DLCPTH (./INCA_libs/irun.lnx86.09.20.nc/cdsrun.lib,1): cds.lib No
such file '/home/chenke/cds.lib' (cds.lib command ignored).
ncvlog: 09.20-p007: (c) Copyright 1995-2009 Cadence Design Systems, Inc.
file: ./tb_intra.v
file: ../rtl/mem/rtl_model/rom_1p.v
file: ../rtl/mem/rtl_model/rf_1p.v
file: ../rtl/mem/rtl_model/rf_2p.v
file: ../rtl/mem/rtl_model/rf_2p_be.v
file: ../rtl/mem/rtl_model/ram_1p.v
file: ../rtl/mem/rtl_model/ram_dp.v
file: ../rtl/mem/rtl_model/ram_dp_be.v
file: ../rtl/intra/ram_frame_row_32x480.v
file: ../rtl/intra/ram_lcu_column_32x64.v
```

- 仿真工具：ncsim
  - 口令：make ncsim
- 仿真界面：
  - Autocheck
  - 自动停止
- 输出波形
  - verdi
- 输出文件

仿真步骤


```
File  Edit  View  Terminal  Tabs  Help
[chenke@vipdev2 sim]$ make ncsim
ncverilog +access+r -l ./dump/tb_intra.log -f file.f -timescale 1ns/1ps
ncverilog: 09.20-p007: (c) Copyright 1995-2009 Cadence Design Systems, Inc.
ncverilog: *W,LOGOPN: log file (./dump/tb_intra.log) could not be opened, contin
uing without it (No such file or directory).
include ../../../../../cds.lib
|
ncverilog: *W,DLCPTH (./INCA_libs/irun.lnx86.09.20.nc/cdsrun.lib,1): cds.lib No
such file '/home/chenke/cds.lib' (cds.lib command ignored).
include ../../../../../cds.lib
|
ncverilog: *W,DLCPTH (./INCA_libs/irun.lnx86.09.20.nc/cdsrun.lib,1): cds.lib No
such file '/home/chenke/cds.lib' (cds.lib command ignored).
ncvlog: 09.20-p007: (c) Copyright 1995-2009 Cadence Design Systems, Inc.
file: ./tb_intra.v
file: ../rtl/mem/rtl_model/rom_1p.v
file: ../rtl/mem/rtl_model/rf_1p.v
file: ../rtl/mem/rtl_model/rf_2p.v
file: ../rtl/mem/rtl_model/rf_2p_be.v
file: ../rtl/mem/rtl_model/ram_1p.v
file: ../rtl/mem/rtl_model/ram_dp.v
file: ../rtl/mem/rtl_model/ram_dp_be.v
file: ../rtl/intra/ram_frame_row_32x480.v
file: ../rtl/intra/ram_lcu_column_32x64.v
```

Make file & file list & testbench

- 仿真工具：ncsim
  - 口令：make ncsim
- 仿真界面：
  - Autocheck
  - 自动停止
- 输出波形
  - verdi
- 输出文件

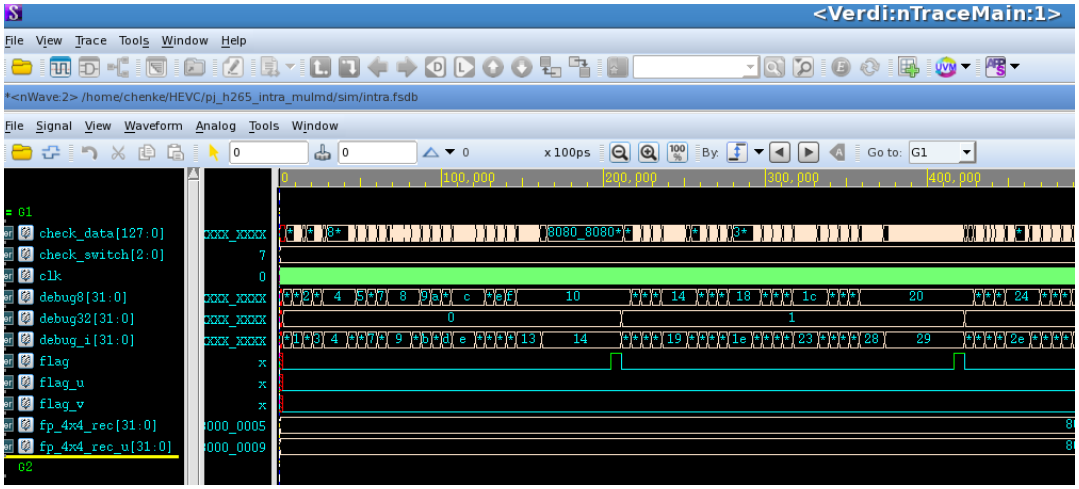
仿真步骤


```
*** CHECK to INTRA ***
                                auto check to pred(yuv) is on
        00000110, fra_no = 00, mb_x = 00, mb_y = 00

      01845325, fra_no = 00, mb_x = 00, mb_y = 03

      01932465, fra_no = 00, mb_x = 01, mb_y = 03

      02018685, fra_no = 00, mb_x = 02, mb_y = 03

      02108585, fra_no = 00, mb_x = 03, mb_y = 03

      02197565, fra_no = 00, mb_x = 04, mb_y = 03

      02286545, fra_no = 00, mb_x = 05, mb_y = 03

      02374605, fra_no = 00, mb_x = 06, mb_y = 03


**check finished! **

Simulation complete via $finish(1) at time 2457705 NS + 0
./tb_intra.v:271         $finish;
ncsim> exit
```

- 仿真工具：ncsim
  - 口令：make ncsim
- 仿真界面：
  - Autocheck
  - 自动停止
- 输出波形
  - verdi
- 输出文件

仿真步骤


```
[chenke@vipdev2 sim]$ verdi &
```



- 仿真工具：ncsim
  - 口令：make ncsim
- 仿真界面：
  - Autocheck
  - 自动停止
- 输出波形
  - verdi
- 输出文件

仿真步骤

Fsdb波形文件

- 仿真工具：ncsim
  - 口令：make ncsim
- 仿真界面：
  - Autocheck
  - 自动停止
- 输出波形
  - verdi
- 输出文件

帧内预测

- 硬件架构
- 硬件接口
- Testbench详解
- 仿真步骤
- 学习资料

学习资料

- 万帅、杨付正.新一代高效视频编码H.265/HEVC：原理、标准与实现
- 陆彦珩.高灵活性HEVC帧内编码器设计
- 沈蔚炜.帧内编码器
- 刘聪. HEVC视频编码器中帧内预测模块的VLSI实现研究

ASIC

专注开源硬件 IP Core

www.openasic.org

谢谢！