实 验 报 告

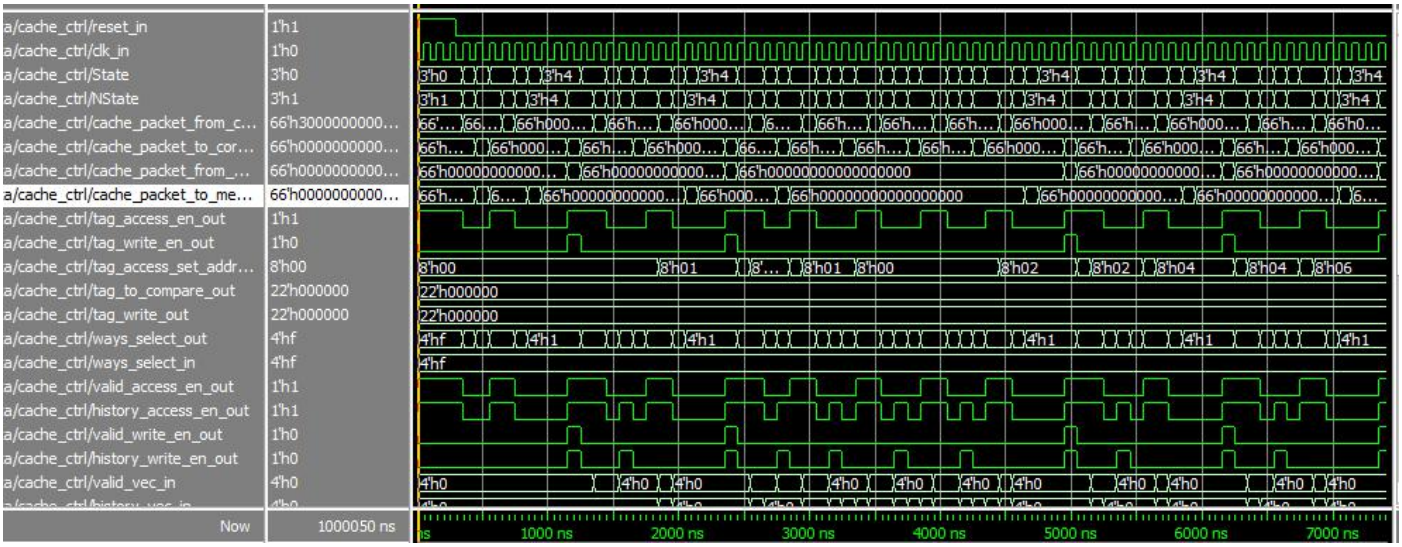
学号：2014K8009929022　姓名：孔静　专业：计算机科学与技术

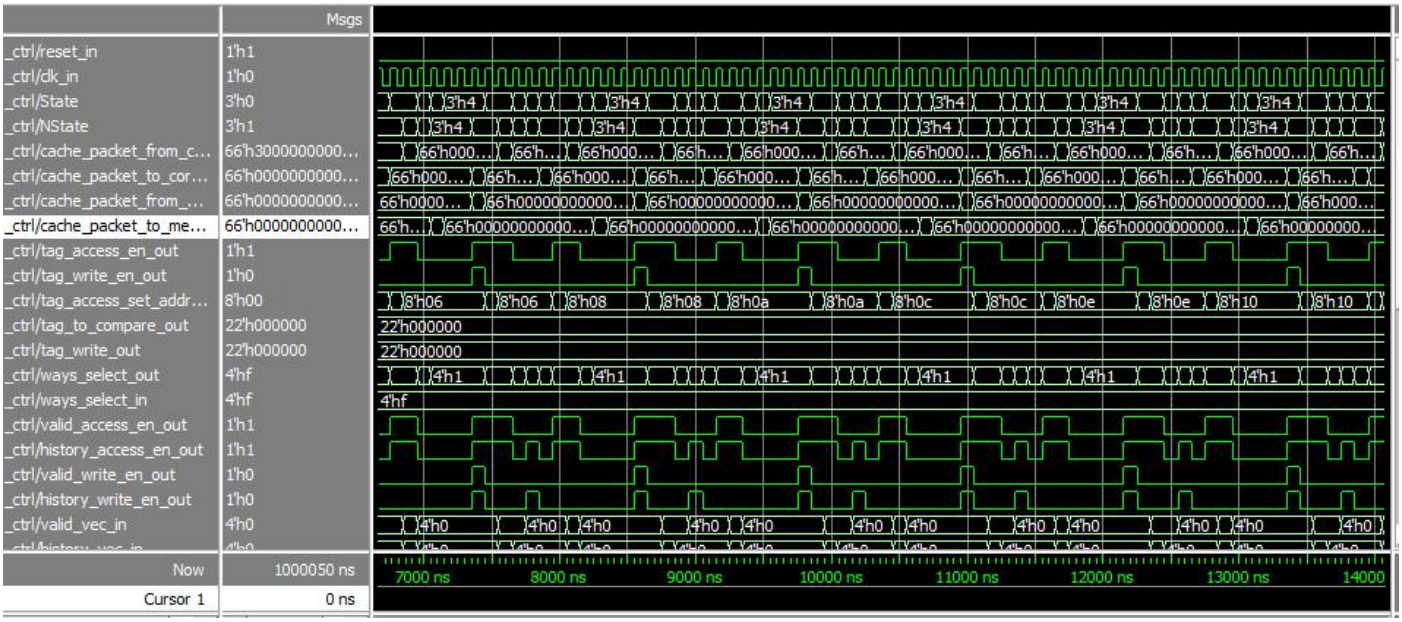
实验序号：5　实验名称：简易高速缓存（Cache）

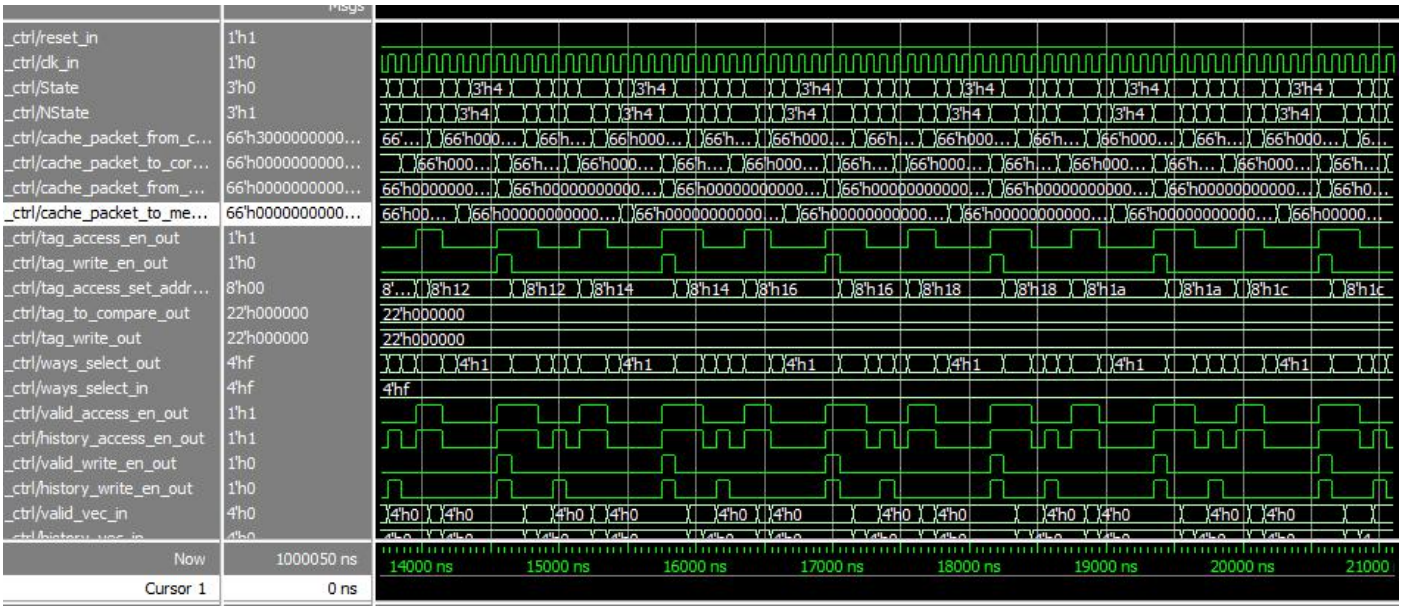
一、　代码以及波形图

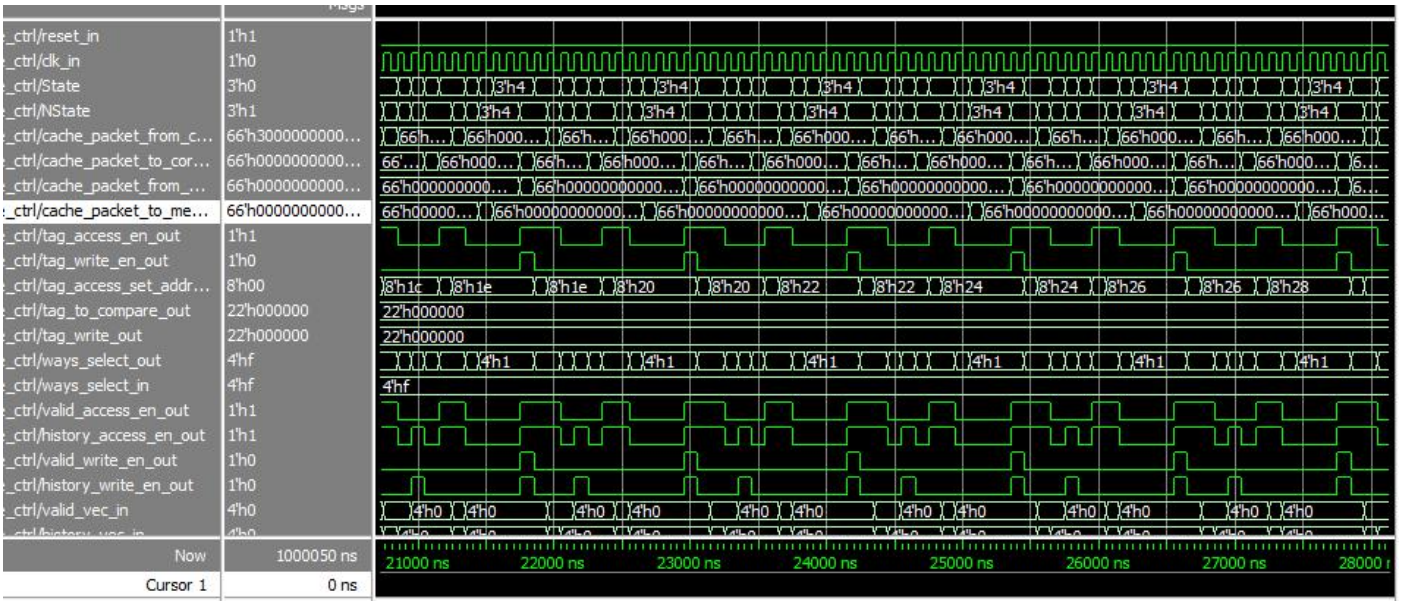
这次实验，根据老师给的 cache.v 以及上课的 ppt 等以及大腿们的指导以及百度的相关资料比如 generate for 的用法，最后完成了这个实验，虽然讲道理说实话的话，我还是不太明白我自己在干什么。即使有大腿们的解释帮助，我还是有点不太懂。=。=再加上考试一波波的来袭，我也懒得再像上次那样搞真值表对每个步骤解释，就让我再水一次吧。

波形图如下~（哦，这次 testbench 用的是传奇大腿提供的，听说好像这是谭大腿写的？不太清楚，反正我用都用了=。=）









代码图~

cache.v(略)

cache_array.v

```
  generate
    genvar i;
    for(i=0;i<NUMBER_WAYS;i=i+1)
    begin : ram_array
      RAM
      #(
        .SINGLE_ELEMENT_SIZE_IN_BITS(SINGLE_ELEMENT_SIZE_IN_BITS),
        .NUMBER_SETS(NUMBER_SETS),
        .SET_PTR_WIDTH_IN_BITS(SET_PTR_WIDTH_IN_BITS)
      )
      ram
      (
        .reset_in(reset_in),
        .clk_in(clk_in),

        .access_en_in(access_en_in&&ways_select_in[i]),
        .write_en_in(write_en_in),
        .access_set_addr_in(access_set_addr_in),
        .write_element_in(write_element_in),
        .read_element_out(read_out[i])
      );
    end
  endgenerate

generate
  genvar j;
  for(j=0;j<NUMBER_WAYS;j=j+1)
  begin
    assign read_full_elements_out[j*SINGLE_ELEMENT_SIZE_I
  end
endgenerate

generate
  genvar k;
  for(k=0;k<NUMBER_WAYS;k=k+1)
  begin
    always@(*)
    begin
      if(compare_in==read_out[k])
        ways_select_out[k]=1;
      else
        ways_select_out[k]=0;
    end
  end
endgenerate

generate
  genvar l;
  for(l=1;l<NUMBER_WAYS;l=l+1)
  begin
    if(l==1)
      assign midor[l-1]=read_out[0]|read_out[1];
    else
      assign midor[l-1]=midor[l-2]|read_out[l];
  end
endgenerate

assign read_element_out=midor[NUMBER_WAYS-2];
```

cache_control.v

```
queue queue(.valid_vec_in(valid_vec_in),.history_vec_in(history_vec_in),.ways_select_in(ways_select_in),.new_ways(new_ways

assign state[0]=(State==3'd0)? 1:0;
assign state[1]=(State==3'd1)? 1:0;
assign state[2]=(State==3'd2)? 1:0;
assign state[3]=(State==3'd3)? 1:0;
assign state[4]=(State==3'd4)? 1:0;

assign packetwrite=state[0]&&cache_packet_from_core_in[`CACHE_PACKET_VALID_POS];
assign write=state[1];

assign tag_access_en_out=state[0]||(state[4]&&cache_packet_from_mem_in[`CACHE_PACKET_VALID_POS]);
assign tag_write_en_out=state[4]&&cache_packet_from_mem_in[`CACHE_PACKET_VALID_POS];
assign tag_to_compare_out=cache_packet_from_core[`CACHE_TAG_POS_HI:`CACHE_TAG_POS_LO];
assign tag_write_out=cache_packet_from_core[`CACHE_TAG_POS_HI:`CACHE_TAG_POS_LO];

assign valid_access_en_out=state[0]||(state[4]&&cache_packet_from_mem_in[`CACHE_PACKET_VALID_POS]);
assign history_access_en_out=state[0]||(state[2]&&hit)||(state[4]&&cache_packet_from_mem_in[`CACHE_PACKET_VALID_POS]);
assign valid_write_en_out=state[4]&&cache_packet_from_mem_in[`CACHE_PACKET_VALID_POS];
assign history_write_en_out=(state[2]&&hit)||(state[4]&&cache_packet_from_mem_in[`CACHE_PACKET_VALID_POS]);

assign data_access_en_out=(state[2]&&hit)||(state[4]&&cache_packet_from_mem_in[`CACHE_PACKET_VALID_POS]);
assign data_write_en_out=(state[2]&&hit&&cache_packet_from_core[`CACHE_PACKET_IS_WRITE_POS])||(state[4]&&cache_packet_fror
assign data_access_set_addr_out=(state[2]||state[4])&&cache_packet_from_core[`CACHE_INDEX_POS_HI:`CACHE_INDEX_POS_LO];


assign valid_write_out=1;
assign history_write_out=state[2]&&hit;

always@(*)
begin
  if(state[2])
    data_write_out=cache_packet_from_core[`CACHE_PACKET_DATA_POS_HI:`CACHE_PACKET_DATA_POS_LO];
  else if(state[4])
    data_write_out=cache_packet_from_mem_in[`CACHE_PACKET_DATA_POS_HI:`CACHE_PACKET_DATA_POS_LO];
  else
    data_write_out=0;

  if(state[0])
    tag_access_set_addr_out=cache_packet_from_core_in[`CACHE_INDEX_POS_HI:`CACHE_INDEX_POS_LO];
  else
    tag_access_set_addr_out=cache_packet_from_core[`CACHE_INDEX_POS_HI:`CACHE_INDEX_POS_LO];

  if(state[0])
    ways_select_out=(1<<(`CACHE_SET_ASSOCIATIVITY))-1;
  else if(state[2]||state[4])
    ways_select_out=ways_select;
  else
    ways_select_out=0;

  if(state[4]&&solution==2'b11)
```

```
if(state[4]&&solution==2'b11)
  history_ways_select_out=(1<<(`CACHE_SET_ASSOCIATIVITY))-1;
else
  history_ways_select_out=ways_select_out;

if(state[2])
  begin
    case({cache_packet_from_core[`CACHE_PACKET_IS_WRITE_POS],hit})
    2'b01:cache_packet_to_mem_out=0;
    default:cache_packet_to_mem_out=cache_packet_from_core;
    endcase
  end
else
    cache_packet_to_mem_out=0;

if(state[2])
  begin
    case(cache_packet_from_core[`CACHE_PACKET_IS_WRITE_POS])
    1'b0:cache_packet_to_core_out=0;
    1'b1:cache_packet_to_core_out=cache_packet_from_core;
    default:cache_packet_to_core_out=0;
    endcase
  end
else if(state[3])
  cache_packet_to_core_out={cache_packet_from_core[(`CACHE_PACKET_WIDTH_IN_BITS)-1:(`CACHE_PACKET_DATA_POS_HI)+1],data_
else if(state[4]&&cache_packet_from_mem_in[`CACHE_PACKET_VALID_POS])
  cache_packet_to_core_out=cache_packet_from_mem_in;
else
  cache_packet_to_core_out=0;
end

always@(*)
begin
  case(State)
  3'd0:
  begin
    if(cache_packet_from_core_in[`CACHE_PACKET_VALID_POS])
      NState=3'd1;
    else
      NState=3'd0;
  end
  3'd1:NState=3'd2;
  3'd2:
  begin
    case({cache_packet_from_core[65],hit})
    2'b10,2'b11:NState=3'd0;
    2'b01:NState=3'd3;
    2'b00:NState=3'd4;
    default:NState=3'd0;
    endcase
  end
  3'd3:NState=3'd0;
  3'd4:
  begin
    if (cache_packet_from_mem_in[`CACHE_PACKET_VALID_POS])
      NState=3'd0;
```

```
      3'd3:NState=3'd0;
      3'd4:
      begin
        if (cache_packet_from_mem_in[`CACHE_PACKET_VALID_POS])
          NState=3'd0;
        else
          NState=3'd4;
      end
      default:NState=3'd0;
      endcase
    end

  always@(posedge reset_in or posedge clk_in)
  begin
    if(reset_in)
      begin
        State<=3'd0;
        cache_packet_from_core<=0;
        ways_select<=0;
        solution<=0;
        hit<=0;
      end
    else
      begin
        State<=NState;
        if(packetwrite)
          cache_packet_from_core<=cache_packet_from_core_in;
        if(write)
          begin
            ways_select<=new_ways;
            solution<=newsolution;
            hit<=|(valid_vec_in&ways_select_in);
          end
      end
  end
```

cache_others.v

queue

```
generate
  genvar i;
  for(i=0;i<(`CACHE_SET_ASSOCIATIVITY);i=i+1)
  begin
    if(i==0)
    begin
      assign new_ways_1[i]=~valid_vec_in[i];
      assign mask1[i]=valid_vec_in[i];
      assign new_ways_2[i]=~history_vec_in[i];
      assign mask2[i]=history_vec_in[i];
    end
    else
    begin
      assign new_ways_1[i]=(~valid_vec_in[i])&&mask1[i-1];
      assign mask1[i]=mask1[i-1]&&(~new_ways_1[i]);
      assign new_ways_2[i]=(~history_vec_in[i])&&mask2[i-1];
      assign mask2[i]=mask2[i-1]&&(~new_ways_2[i]);
    end
  end
endgenerate
```

```
always@(*)
begin
  if(|(valid_vec_in&ways_select_in))
  begin
    solution=2'b00;
    new_ways=valid_vec_in&ways_select_in;
  end
  else if((&(valid_vec_in))==0)
  begin
    solution=2'b01;
    new_ways=new_ways_1;
  end
  else if((&(history_vec_in))==0)
  begin
    solution=2'b10;
    new_ways=new_ways_2;
  end
  else
  begin
    solution=2'b11;
    new_ways=1;
  end
end
```

ram

```
 generate
   genvar i;
   for(i=0;i<NUMBER_SETS;i=i+1)
   begin
     always@(posedge clk_in or posedge reset_in)
     begin
       if(reset_in)
       begin
         ram[i]<=0;
       end
       else if(access_en_in&&write_en_in&&i==access_set_addr_in)
       begin
         ram[i]<=write_element_in;
       end
     end
   end
 endgenerate

 always@(posedge clk_in or posedge reset_in)
 begin
   if (reset_in)
   begin
     read_element_out<=0;
   end
   else if(access_en_in)
   begin
     read_element_out<=ram[access_set_addr_in];
   end
   else
   begin
     read_element_out<=0;
    end
 end
```

二、 问题

这次写的错误，包括格式错误啊，拼写拼错啊，全部略去，不想回忆，实在是惨痛的改 bug 经历。看了半天没错最后发现只是这种低级错误，实在是不忍直视，浪费我青春啊。

Problem1

问题：刚开始打算写的时候，说要拿状态机写，但是根本不知道怎么下手。

解决：大腿给出意见，按照 cache 的不同工作状态，写出相应状态，并弄出相应真值表。

Problem2

问题：完全不知道应该如何调试，以前大家设置的东西都一样，写法也差不多，所以直接拿大腿的波形图跟我的波形图比对，就知道自己错在哪里了，这次不知道自己怎么是错怎么是对。

解决：耐心下来，一个个状态和相应指令看过去，分析是否正确出错。

三、 对于此次实验的心得、感受和建议

=。=没有心得，只知道这次实验结束，我不想再接触这什么鬼 verilog，调试 bug 真的是太痛苦了。往往写好不需要太久，但是调 bug 却可以调好几倍的时间。一想到今后还有更多的代码要写，更多的 bug 要调，我选择死亡(ノ｀□′)ノ ┻━┻。

再次感谢帮助我的大腿们。没怎么帮过他们，他们却一直不厌其烦的帮我解决问题，唉，还是我太弱了，强一点就可以帮他们忙，毕竟只是得到没有付出，感觉良心不安啊。

心累，没有建议。