# 2016

一、简述 interface 的作用及如何用 interface将模块仅连接到本模块用到的信号上。(5分)

P219 P226

interface允许多信号合成一组由一个端口表示，只需在一个地方对组成接口的信号进行声明，每个用到这些信号的模块只需在一个地方对组成接口的信号进行声明，这样可以克服verilog模块需要大量冗余声明的缺点。同时在接口中可以定义通信协议，在接口中也可以直接建立协议校验和其他验证程序。

可以用modport来定义接口信号的不同接入方式，modport可以定义某一模块用到接口中哪些信号，且定义了从模块角度来看接口信号的端口方向。

二、给出以综合为目的的模块划分原则。(5分）

PPT的21页

* 1. 没有组合逻辑跨越层次结构边界；
  2. 将层次结构边界放在寄存器输出的地方；
  3. 合理划分每一块的大小，以获得较短的综合运行时间；
  4. 将Logic Core、pads、clocks、JTAG分开综合。

三、简述unique case与综合指令 parallel case的区别。(5分)

P158

Unique case包含了parallel\_case的附注功能，而且还附带了语法检查。对于parallel\_case附注，当多个条件选项成立时，parallel\_case会强制并行求值，这样可能会导致同时执行多条case分支语句；而对于unique\_case来说，若设计者对条件语句的并行、完整的假设是错误的，unique case就会产生运行期警告，parallel\_case就不会对条件选项做任何检查。

1. 举例说明用结构体作为模块端口传递数据时的规则。(5分)

P79

1.结构体必须首先使用typedef定义成用户自定义类型，然后才允许模块或者接口的端口声明为结构体类型；

2.当一个非压缩结构体通过端口传递时，端口两边连接的必须是同一种类型的用户自定义结构体；

3.压缩结构体可以当成向量作为端口传递。

五、举例说明用缺省基类型的枚举类型设计状态机的潜在问题，并简述设计状态机的一般原则。

P164 P173 三段式

typedef enum{RED,GREEN,YELLOW} traffic\_t

traffic\_t State,Next；

Always\_ff@(posedge clock,negedge resetN)

If(!resetN) State <= RED;

Else State<=Next;

Always\_comb begin

Next = State; //下面每个分支的缺省状态

Unique case (state)

RED : if(sensor) Next = GREEN;

GREEN:if(green\_countdown == 0) Next = YELLOW;

YELLOW:if(yellow\_countdown==0) Next = RED;

Endcase

End

Always\_comb begin:

{green\_light,yellow\_light,red\_light} = 3’b000;

Unique case(state)

RED : red\_light = 1’b1;

GREEN:green\_light = 1’b1;

YELLOW:yellow\_light = 1’b1;

Endcase

end

1.默认的枚举基类int可能不能精确反应仿真的硬件行为，int类型是一个32位的两态类型，而上例的实际硬件只需要一个两位或者三位的向量，并且实际硬件实现的门级模型将具有四态语义；

2.缺省的两态类型仿真时的默认初始值会掩盖仿真问题；仿真刚开始时，两态数据类型的缺省值是0而不是X，而复位操作通常是将大部分变量清0，这样一来复位逻辑有缺陷，模型好像仍然是被复位了。

3.缺省类型的复位+使用always@来书写敏感表容易将状态机的状态锁定。

4.枚举标签的默认值也可能导致设计的RTL仿真和门级实现之间不一致，这样会难以比较综合前后模型功能。

状态机设计的一般原则：

1. 采用三段式：状态更新；下一状态计算；输出逻辑。
2. 将unique case和枚举类型的四态基类结合起来使用，以检查复位功能和防止 状态锁定。Always\_comb

六、举例说明always\_comb与always @\*的区别。（10分)

P114

1.Always@\*推断出的敏感表只针对那些被always块直接读取的信号，而对过程块调用的函数中读取的信号不能推断为敏感量；always\_comb过程块对块内读取的信号和块内调用的函数读取的信号都敏感，这样编写函数时就不需要形式参数了。例子如下

always@\* begin →推断出@（data）

A1 = data <<1;

B1 = decode();

End

Always\_comb begin →推断出@(data sel c d e)

A2 = data<<1;

B2=decode();

End

Function decode;

Begin

Case(sel)

2’b01 : decode = d|e;

2’b10: decode = d&e;

Default : decode = 1;

Endcase

End

endfunction

2.Always\_comb在零时刻自动执行保证了输出与输入值的一致性，与之相比，@\*只有在敏感信号发生变化时才触发。（5中状态机状态锁定的例子）。

3.@\*没有组合逻辑语义，也可能会因为不正确使用而导致不可综合。

1. 给出考研成绩的数据结构:要求能够按性别、单科成绩、总成绩分别排序。（10分)

typedef enum {male,female} gender\_t;

typedef struct {

sex\_t SEX;

int MATH;

int POLITICS;

int ENGLISH;

int MAJOR;

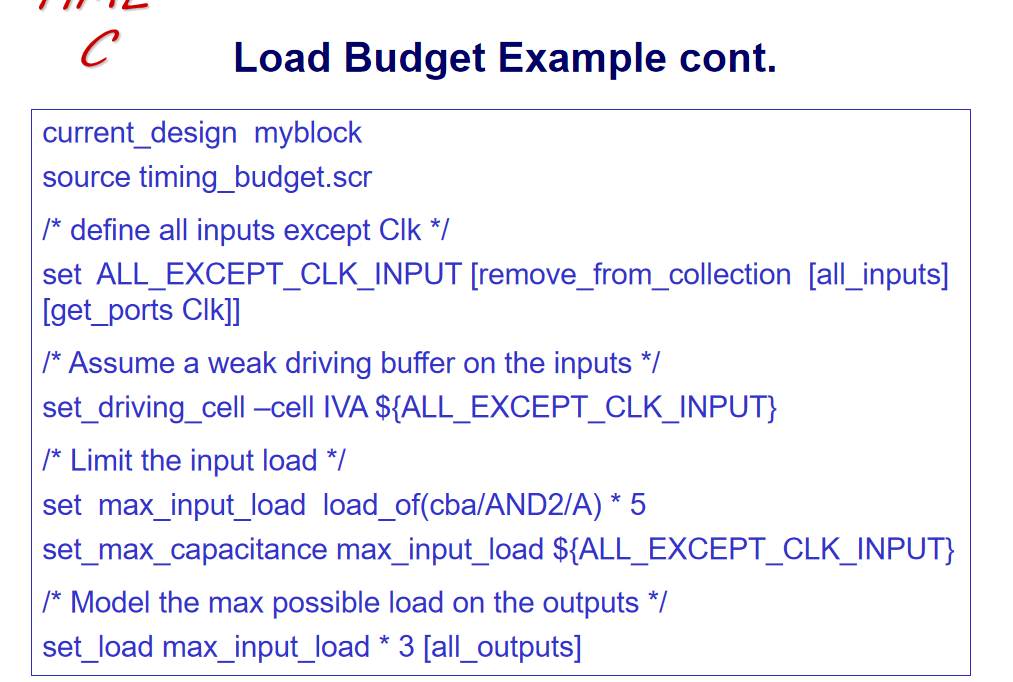
int SUM;

}exam\_grade\_t;

八、给出一个顶层名为CPU\_TOP的模块的综合约束，自己命名如目标库，线载模型等必须用到的名称。(10分)

PPT的14页、48页、51页、55页、63页、65页、71页、72页  **总结在74页（总结里缺少设置目标库）**





setup tcl

set target\_library {smic\_018.db}

set link\_library {\* smic\_018.db dw\_foundation.sldb}

set synthetic\_library {dw\_foundation.sldb}

set symbol\_library {smic018.sdb}

set search\_path [list . ${smic\_std\_cell}]

current\_design CPU\_TOP

create \_clock -period 10 [get\_ports Clk]

set\_dont\_touch\_network [get\_clocks Clk]

set\_input\_delay -max 6.0 -clock Clk [all-inputs]

set\_output\_delay -max 6.0 -clock Clk [all-ouputs]

set\_max\_area 0

set\_load load\_of(smic018/AN2/A) [all-outputs]

set\_driving\_cell -lib\_cell”ND2” [all-inputs]

set\_operating\_conditions -max “typ\_-40\_4.50”

set\_wire\_load tc6a120m2

set\_max\_fanout 3

set\_max\_transition 0.4

set\_max\_capacitance 10

report\_port -verbose

report\_design

report\_clock

write\_script

# 2017

一、简述systemVeriloo中接口的优势，并用接口定义一个简单的总线，并用modport对不同类型设备使用的端口进行归类。15分

接口的优势：

interface允许多信号合成一组由一个端口表示，只需在一个地方对组成接口的信号进行声明，每个用到这些信号的模块只需在一个地方对组成接口的信号进行声明，这样可以克服verilog模块需要大量冗余声明的缺点。同时在接口中可以定义通信协议，在接口中也可以直接建立协议校验和其他验证程序。

可以用modport来定义接口信号的不同接入方式，modport可以定义某一模块用到接口中哪些信号，且定义了从模块角度来看接口信号的端口方向。

interface bus\_if(

input clk,

input resetn

);

logic [31:0] ADDR;

logic [31:0] WDATA;

logic [31:0] RDATA;

logic HBUSREQ [1:0];

logic HGRANT [1:0];

logic HSEL [1:0];

logic HREDY;

modport master();

modport slave();

modport arbiter();

modport decoder();

endinterface

二、以顶层设计名为h264,使田的工艺库为smic018,综合工具为DesionCompiler为例，给出从时序、工作环境、面积和设计规则等方面对h264的综合约束文件,对自定义的变量给出必要的说明。

setup tcl

set target\_library {smic\_018.db}

set link\_library {\* smic\_018.db dw\_foundation.sldb}

set synthetic\_library {dw\_foundation.sldb}

set symbol\_library {smic018.sdb}

set search\_path [list . ${smic\_std\_cell}]

current\_design h264

create \_clock -period 10 [get\_ports Clk] //时钟周期

set\_dont\_touch\_network [get\_clocks Clk]

set\_input\_delay -max 6.0 -clock Clk [all-inputs] //外部占60%

set\_output\_delay -max 6.0 -clock Clk [all-ouputs]

set\_max\_area 0 //按面积最优

set\_load load\_of(smic018/AN2/A) [all-outputs] //取sim018库中AN2这cell的A pin的负载作为标准，复制到所有输出端口上

set\_driving\_cell -lib\_cell”ND2” [all-inputs] //取ND2的驱动能力作为所有输入端口的驱动单元能力

set\_operating\_conditions -max “typ\_-40\_4.50” // 工作环境为 -40摄氏度，电源电压 4.5V

set\_wire\_load tc6a120m2 //线载模型为tc6a120m2

set\_max\_fanout 3 //最多驱动不超过3个模块

set\_max\_transition 0.4 //时钟上升过度时间设置为0.4

set\_max\_capacitance 10 //设置输入端口的输入电容

report\_port -verbose

report\_design

report\_clock

write\_script

三、给出大学生数学建模竞赛的数据结构,内容包括性别、学生类别（博士生、硕士生、本科生)、及分数(开题报告分数模型报告分数、答辩分新、总分数)。15分

typedef enum {male,female} gender\_t;

typedef enum{doctor,post\_graduate,under\_graduate} education\_t;

typedef struct {

string name;

gender\_t gender;

education\_t eductation;

int previous\_report\_score;

int model\_report\_score;

int presentation\_report\_score;

int tota\_score;

}MCM\_info\_t;

四、简述集成电路设计时模块划分的一般原则。5分

* 1. 没有组合逻辑跨越层次结构边界；
  2. 将层次结构边界放在寄存器输出的地方；
  3. 合理划分每一块的大小，以获得较短的综合运行时间；
  4. 将Logic Core、pads、clocks、JTAG以及异步电路分开综合。

1. 举例说明always comb推断如何推断出函数/任务中的信号为敏感表。10分

P110

Always\_comb begin →推断出@(data sel c d e)

A2 = data<<1;

B2=decode();

End

Function decode;

Begin

Case(sel)

2’b01 : decode = d|e;

2’b10: decode = d&e;

Default : decode = 1;

Endcase

End

endfunction

如上例，在always过程块调用的无参数function内部所需要的一系列信号，也会被always\_comb推断出来，只被函数赋值和读取的临时变量除外。

六、举例说明使用unique case时，当条件项不全时,综合时会不会产生锁存器。10分

P156 P157

不会

logic [2:0] opcode;

always\_comb

unique case(opcode)

3’b000: y = a+b;

3’b001: y = a-b;

3’b010: y = a\*b;

3’b100: y = a/b;

endcase

如果opcode的值是3 5 6 7 那么会产生运行期警告，这样就保证了在case语句求值时，有且只有一个分支执行，即case语句由组合逻辑实现，不会产生锁存器。

1. 给出 systemVerilog对函数的改变中哪些是对设计真正有利的改善。5分 P119
   1. systemverilog增加了一条return语句，return的优先级高于返回函数名的值；return可以在任务和函数结束前提前返回，简化代码，使得执行过程更直观。
   2. 可以定义一个无返回值的空函数，调用方式和任务一样；空函数克服了函数不能调用任务，这改善了难以在复杂函数中添加代码结构的缺陷。空函数可以有inout和output的形式参数。同时空函数和空任务在非完整代码中预留了空间，有利于top down设计。
   3. systemverilog中的任务和函数可以用名称传递参数。
   4. 函数可以有任意个输出和返回值，极大地扩展了函数的建模范围。
2. 用systemVerilog描述以下功能，数据输入为8bit端口，输入的数据要填入数据包中，该数据包由8bit命令，32bit数据，24bit地址，8bit校验位，其中效验位由命令、数据、地址生成，且每8bit生成1bit的校验位。20分

方法一：

typedef struct packed{

logic [7:0] check;

logic [7:0] cmd;

logic [31:0] data;

logic [23:0] addr;

}packet\_t;

function void fill\_packet(

input logic [7:0] data\_in [0:7],

output packet\_t data\_out

);

union packed{

packet\_t data\_out;

logic [7:0][8:0]bytes;

}dreg;

for(int i = 0;i <= 7;i++) begin

dreg.bytes [(8\*i) + :8] = data\_in[i];

dreg.data\_out.check[i] = ^data\_in[i];

end

endfunction

方法二：

typedef struct packed{

logic [7:0] check;

logic [7:0] cmd;

logic [31:0] data;

logic [23:0] addr;

}packet\_t;

function void fill\_packet(

input logic [7:0] data\_in [0:7],

output packet\_t data\_out

);

for(int i = 0;i <= 7;i++) begin

data\_out [(8\*i) + :8] = data\_in[i];

data\_out.check[i] = ^data\_in[i];

end

endfunction

# 2015

1. 分析并给出 while(k+t<LIMIT)和 while(+tj< LIMIT)循环的循环次数(5分)while(k++<LIMIT)的循环次数是LIMIT+1次；while（++j<LIMIT）的循环次数是LIMIT次。

k++是先比较然后再加1：k从0开始，循环第一次的时候k为0，循环第LIMIT次的时候k是LIMIT-1，此时仍然满足判断条件，所以仍然继续循环到LIMIT+1次，此时k为LIMIT，不满足循环条件，退出循环。

++j是先加1再比较：j从0开始，循环第一次的时候j为1，循环第LIMIT-1次的时候，j是LIMIT-1，此时仍然满足判断条件，所以仍然继续循环到LIMIT次，此时j为LIMIT，不满足循环条件，退出循环。

二、给出 always\_comb 与 always@\*的异同。(5分)

P114

同：@\*和comb都是过程块的开始，都可以推断出敏感表，都可以更方便地描述复杂的组合逻辑过程快。

异：

1. comb不仅仅推断出组合逻辑敏感表，并且严格限制了其他过程块对相同变量的赋值，这个限制确保了组合行为的正确性。
2. comb在零时刻自动执行的特性保证了输出与输入值行为的一致性，与之相比，@\*敏感表只能是至少有一个敏感信号发生变化时才触发。
3. @\*推断出的敏感表只针对那些被always块直接读取的信号，而对过程块调用的函数中读取的信号不能推断为敏感量。 comb过程块对块内读取的信号和块内调用的函数读取的信号都敏感。

三、举例说明压缩结构体的操作。(5分)

P77 P78

packed关键可以显示地声明一个压缩结构体。

压缩结构体的操作：由于压缩结构体以向量形式存储，对整个结构体的操作也是以向量形式，因此对向量的算数操作、逻辑操作以及任何其他操作都可以用于压缩结构体。如下面对压缩结构体进行向量的移位的操作和，以及截取向量中某几位进行赋值。

typedef struct packed{

logic valid;

logic[7:0] tag;

logic [31:0] data;

}data\_word\_t;

data\_word\_t packet\_in,packet\_out;

always@(posedge clock)

packet\_out <= packet\_in <<2;

packet\_out[39:32] = 8’hf0;

1. 给出systemverilog对函数的不影响设计可移植性、可读性和复用性的重要扩展。 P119
   1. systemverilog增加了一条return语句，return的优先级高于返回函数名的值；return可以在任务和函数结束前提前返回，简化代码，使得执行过程更直观。
   2. 可以定义一个无返回值的空函数，调用方式和任务一样；空函数克服了函数不能调用任务，这改善了难以在复杂函数中添加代码结构的缺陷。空函数可以有inout和output的形式参数。同时空函数和空任务在非完整代码中预留了空间，有利于top down设计。
   3. systemverilog中的任务和函数可以用名称传递参数。
   4. 函数可以有任意个输出和返回值，极大地扩展了函数的建模范围。
2. 举例说明有无关通配符相等操作符（?）的操作结果。(5分) P137

在通配符操作符==？的使用中，右边操作数中的一位逻辑X或逻辑Z值都可以被看做一个通配符，而这个通配符可以和另一个操作数中相应位的任何数值相匹配。

a = 0101 b= 010X a ==? b 结果为真

1. 举例说明如何应用结构体作为模块端口。(5分)P79
2. 结构体必须首先使用typedef定义成用户自定义类型，然后才允许模块或者接口的端口声明为结构体类型；
3. 当一个非压缩结构体通过端口传递时，端口两边连接的必须是同一种类型的用户自 定义结构体；
4. 压缩结构体可以当成向量作为端口传递。如何端口两端压缩结构体大小不同，会按照verilog的标准以向量长度不匹配的情况进行处理

typedef struct{

logic [31:0] a,b;

logic[23:0] address;

logic error;

}instr\_t

module alu (input instr\_t IW, output result);

……

1. 用if语句描述一个与条件选项顺序无关的4选一选择器。(5分)

P160

module mux\_41(input logic [3:0]sel,

input logic a,

input logic b,

input logic c,

output result);

always\_comb begin

unique if(sel[0]) result =a;

else if(sel[1]) result=b;

else if(sel[2]) result=c;

end

endmodule

1. 如何用interface进行硬件设计?modport及接口方法作用是什么?（5分)

九、利用结构、联合、枚举、数组等知识描述体操比赛运动员的数据结构:要求能够按性别、单项成绩、全能成绩分别排序。(10分)P219 P222 P232

1. 接口可以像模块那样拥有端口，使得时钟信号和复位信号可以封装到接口中，称为接口所描述的一组信号的一部分。
2. 接口名称可以在两种情况下引用，一是在模块端口中，二是在接口实例中。接口作为模块端口时，不需要考虑文件次序相关性。
3. 接口可以嵌套在模块内，作为局部接口；或者定义在全局，在设计的任何层次、任何模块中当作端口来使用。
4. 模块端口可以显示的声明为特定接口类型，也可以用关键字interface定义成通用接口端口类型。
5. 接口的端口也可以定义为一个接口。
6. 内部信号的引用方式：bus.address

modport的作用：

modport可以被用来定义接口信号的不同接入方式，modport可以定义某一具体模块用到接口中哪些信号，且定义了从模块角度来看这些具体接口信号的端口方向。使用哪种modport可以在模块实例的接口连接中说明或者在模块定义的端口声明时说明。

接口方法的作用：

使用接口方法，模块之间的通信的细节可以转移到接口内描述，从而，与模块间通信相关的代码就不需要再每个模块中重复编写，作为接口方法的代码只需要编写一次，并被每个通过接口连接的模块所共享。在各个模块中都可调用接口的方法，而不必在模块内执行通信协议的功能。这样，接口不仅可以用来封装连接模块的数据，还可以用来封装模块间的通信协议。

1. 利用结构、联合、枚举、数组等知识描述体操比赛运动员的数据结构:要求能够按性别、单项成绩、全能成绩分别排序。(10分)

typedef enum{male female} gender\_t;

typedef struct {

string name;

gender\_t gender;

int single\_subject;

int all\_subject;

}sporter\_t;

十、工艺库smic018包括以下内容:工作环境为typical,线载模型为 tc6a10k,最小驱动的反向器INA的输入端口为IA，最大转换时间为12时间单位;利用以上参数给出一个名为RISC的时钟信号端口为ck的设计的综合约束,要求包括时序、面积和环境三方面约束。(10分)setup tcl

set target\_library {smic\_018.db}

set link\_library {\* smic\_018.db dw\_foundation.sldb}

set synthetic\_library {dw\_foundation.sldb}

set symbol\_library {smic018.sdb}

set search\_path [list . ${smic\_std\_cell}]

current\_design RISC

create \_clock -period 10 [get\_ports clk]

set\_dont\_touch\_network [get\_clocks clk]

set\_input\_delay -max 6.0 -clock clk [all-inputs]

set\_output\_delay -max 6.0 -clock clk [all-ouputs]

set\_max\_area 0

set\_load load\_of(smic018/AN2/A) [all-outputs]

set\_driving\_cell -lib\_cell”IA” [all-inputs] //最小驱动反相器 输入端口

set\_operating\_conditions typical //工作环境为typical

set\_wire\_load tc6a10k //线负载为tc6a10k

set\_max\_fanout 3

set\_max\_transition 12 //最大转换时间

set\_max\_capacitance 10

report\_port -verbose

report\_design

report\_clock

write\_script

# 2014

一、对本课程印象最深的是什么？学习了哪些知识？还希望学习哪些内容，有什么建议?(5分)印象最深的是作业多。希望增加一些验证和后端的知识。

1. 给出逻辑综合5个命令,#说明其用法

set\_max\_area 对面积进行约束，后面可以接一个大致估计的面积，或者接0表示综合 时使面积尽量小

set\_input\_delay/set\_output\_delay 对时序进行约束，设置输入端口/输出端口外部逻辑 所占的时间是多少，从而控制输入端口相连的内部组合逻辑的时间，以满足时序收敛。 后接一个时间单位，一般占时钟周期的60%。然后再用[get\_ports XX]说明设置到哪个端 口。同时还要用-clock clk指明所对应的是哪个时钟

set\_wire\_load 设置连线的负载，后接工艺库里的某一线网负载型号。

set\_load 设置输出端口的负载，后面一般接load\_of命令+某一工艺库/cell/pin的名称， 以获得该pin的负载，然后再用[get\_ports XX]说明设置到哪个端口。

set\_dont\_touch\_network 在创建时钟时使用，通知综合器在生成时钟的时候不要加入 大量的buffer进行驱动加强。后面一般接 [get\_clocks clk] 来说明对应的是哪个时钟。

1. 举例说明unique修饰符的用法和使用unique修饰符的好处,(5分)

P154

unique case指的是分支条件中有且只有一个分支满足判断条件；

unique if语句指的是在综合的时候要去掉优先级编码。

unique能防止无意中产生所存逻辑，若多条分支同时触发或者没有分支被触发，都会 产生运行期警告，这样这种判定序列描述的逻辑会由不含锁存器的带优先编码的组合逻 辑实现。

1. 列举不少于5条systemverilog相对于verilog不影响设计可移植性、可读性和复用性的重要扩展。
   1. 枚举类型的加入
   2. 结构体类型的加入
   3. 函数和任务增加了return语句
   4. interface的加入以及modport和接口方法的加入
   5. always\_comb、always\_ff等systemverilog特有的过程快的加入。
2. 利用结构、联合、枚举数组等知识描述研究生课程成绩的数据结构，要求有合理的层次,且能方便按总成绩排序。（10分》

typedef enum {male female } gender\_t

typedef struct{

string name;

gender\_t gender;

int math;

int soc;

int vlsi;

int sv;

int total;

}grades\_t;

七、设计一个不少于4个状态的状态机及其验证程序。(5分见作业

module FSM(input logic clk,rst,

output logic [3:0]result);

typedef enum{S0,S1,S2,S3} state\_t;

state\_t state,next;

always\_ff@(posedge clk) begin

if(!rst) state <= S0;

else state <= next;

end

always\_comb begin

case(state)

S0:next = S1;

S1:next = S2;

S2:next = S3;

S3:next = S1;

default :next = S1;

endcase

end

always\_comb begin

case(state)

S0:result = 4’b0001;

S1:result = 4’b0010;

S2:result = 4’b0100;

S3:result = 4’b1000;

default : result = 4’b0000;

endcase

end

endmodule