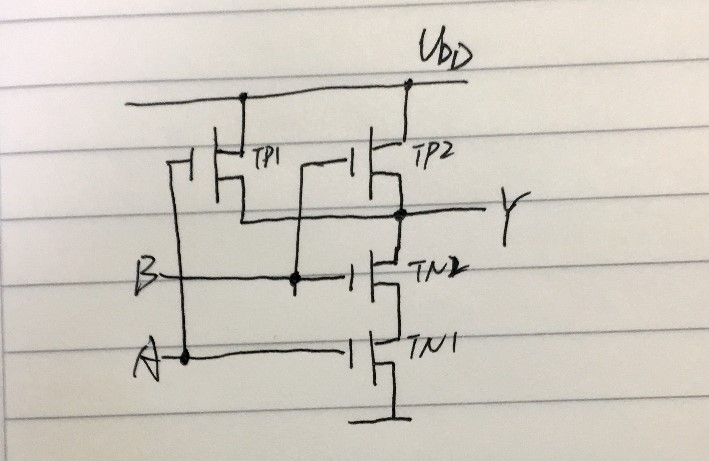
**吴万强—数字IC设计笔试答案**

公共题目：

1. D
2. 10个瓶子依次从1到10标号，然后从1号瓶里取1个药片，2号里取2个，……10号里取10个，然后把这55个药片放在一起称出实际重量，再减去55个标准药片的重量，最后用多出的重量除以超重药片多出的重量，这样得数是几就说明几吃瓶子是超重的药瓶。

数字IC题目：

1、与非门cmos电路



2、与非门真值表

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **Y** | **仿真结果** |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 0 | x | x | 1 |
| 0 | z | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | x | 1 | x |
| 1 | z | 0 | x |
| x | 0 | x | 1 |
| x | 1 | x | x |
| x | x | x | x |
| x | z | x | x |
| z | 0 | 1 | 1 |
| z | 1 | 0 | x |
| z | x | x | x |
| z | z | 0 | x |

3、在ASIC代码中采用异步复位、同步释放的方式。

采用异步复位的原因是：同步系统，一般都使用异步复位，这是因为同步复位的电路实现比异步复位的电路实现要浪费更多电路资源；

采用同步释放的原因是：采用第二级寄存器输出作为全局复位信号有利于减少亚稳态的出现。

4、

（1）二进制编码和格雷码最大的区别是在数字发生变化时二进制编码可能几个bit为都会发生变化，而格雷码始终只会有一个bit位发生变化；因此格雷码更适合于FIFO编码，由于变化的位数少所以跨时钟域传输时可减少亚稳态的发生，且由于翻转较少所以功耗也会更低。

（2）

（3）

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **十进制** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **二进制码** | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
| **格雷码** | 0000 | 0001 | 0011 | 0010 | 0110 | 0111 | 0101 | 0100 |
| **十进制** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |
| **二进制码** | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| **格雷码** | 1100 | 1101 | 1111 | 1110 | 1010 | 1011 | 1001 | 1000 |

（4）对于fifo深度不是2^N时，在编码时转格雷时增加2的偏移，格雷转2进制时减去2的偏移。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **十进制** | **2** | **3** | **4** | **5** | **6** | **7** |
| **二进制码** | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
| **格雷码** | 0011 | 0010 | 0110 | 0111 | 0101 | 0100 |
| **十进制** | **8** | **9** | **10** | **11** | **12** | **13** |
| **二进制码** | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 |
| **格雷码** | 1100 | 1101 | 1111 | 1110 | 1010 | 1011 |

5、第9行output [7:0] data\_out后的 , 应删除；

端口声明完成后括号外缺少;

第二个always块中的敏感信号列表应改为\*比较好，该块下面if语句没有else分支容易产生锁存器应加上else分支；

还有一些赋值符号问题。

修改如下：

module test (

input clk,

input [3:0] addr,

input ren,

input rst\_n,

input [7:0] wdata,

input wen,

output reg [7:0] data\_out

);

reg [7:0] mem\_name [7:0];

reg [3:0] i;

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

if (rst\_n) begin

for(i=0;i<=7;i=i+1) begin

mem\_name[i] <= 0;

end

end

else if (wen)

mem\_name[addr] <= wdata;

else

;

end

always @ (\*)

if (ren)

data\_out = mem\_name[addr];

else

data\_out = data\_out;

endmodule

6、module seqsum (

input clk,

input rst\_n,

input [31:0] a,

input [31:0] b,

output [31:0] y

);

// 时序逻辑实现

wire add\_cnt;

wire end\_cnt;

reg [31:0] cnt;

reg [31:0] y1;

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

if(!rst\_n) begin

cnt <= 0;

end

else if(add\_cnt) begin

cnt <= cnt + 2;

end

else if(end\_cnt) begin

cnt <= cnt;

end

end

assign add\_cnt = a < b && cnt <= b - a; // 由于原函数i从a开始故可将b减去a，则与从a开始计数相同

assign end\_cnt = cnt > b - a;

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

if(!rst\_n) begin

y1 <= 0;

end

else if(add\_cnt) begin

y1 <= y1 + a + cnt; // 与原函数中y=y+i相同，i = a + cnt。

end

end

assign y = end\_cnt?y1:0;

endmodule

7、问题1：

1.黑色箭头代表事件可能发生咋不同的时间。

2.在仿真的各个阶段会发生的事情不同，所以事件的多少也不同。