



中国半导体集成电路高新技术教育培训

联发科笔试题

讲师：xxx

时间：xxxx年xx月xx日

关于联发科技:

联发科技为全球第四大无晶圆半导体公司, 我们所研发的芯片一年驱动超过 15 亿台智能终端设备

核心业务:

移动通信、智能家居、车用电子

工作地:

北京, 上海, 深圳, 合肥, 成都, 武汉



芯片设计验证高级工程师

岗位职责:

1. 负责手机基带芯片设计、验证与整合, 包括modem CPU, 时钟, 总线, 低功耗控制, 外设接口等模块的设计验证以及4G/5G/WIFI基带数据处理模块的验证工作;
2. 与设计, 通信算法和软件团队紧密合作, 完成整个基带数据链的完整验证;
3. 参与所有数字前端设计流程, 包括综合, 时序收敛及所有流片前的规则检查, 提升设计质量, 包括面积, 功耗, 时序及测试覆盖率;

任职要求:

1. 硕士学历, 有数字前端设计、实现或验证的项目经验;
2. 精通Verilog HDL硬件描述语言并具备扎实的数字电路基础;
3. 了解ASIC或FPGA前端设计流程及相关EDA工具, 包括电路综合、时序分析等;
4. 具备以下任一经验者尤佳: 脚本语言运用经验(Perl, Python...), SystemVerilog/UVM验证经验, 熟悉4G/5G/WIFI 通信标准。



芯片设计高级工程师

岗位职责:

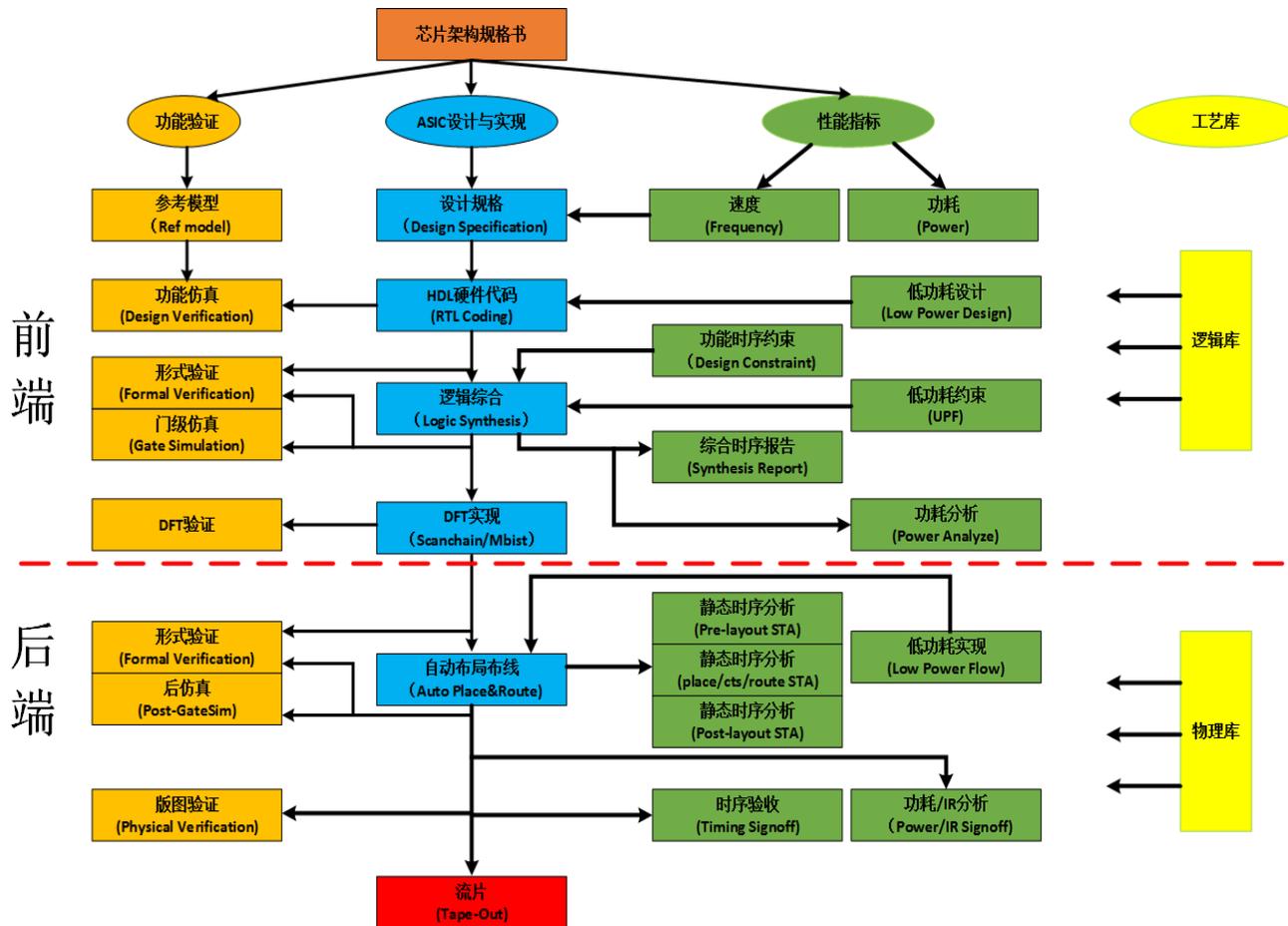
1. 芯片验证/DFT先进技术和流程开发, 以及项目导入;
2. CPU等关键IP, 超大规模SoC的验证/DFT项目执行;
3. 开发EDA自动化软件, 利用大数据等技术改善设计质量及效率;

任职要求:

1. 微电子、计算机、电子工程、通信工程、自动化等相关专业, 硕士及以上学位;
2. 熟悉ASIC前端设计的流程, 具有RTL、FPGA相关经验;
3. 开放, 友善, 有效的沟通方式和技巧, 较强的学习能力和解决问题能力;
4. 加分项: SystemVerilog/UVM/CPU/DFT等相关经验。



1.ASIC流程,说出五个以上环节; Verilog 说出五个以上keyword。

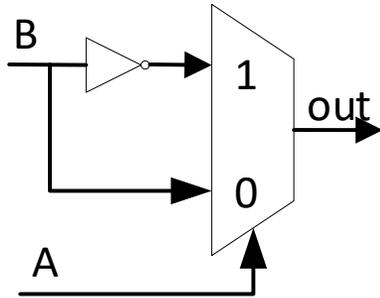


关键字:

Parameter;always;assign;input;
 output;generate;module;
 Fork join;case;function



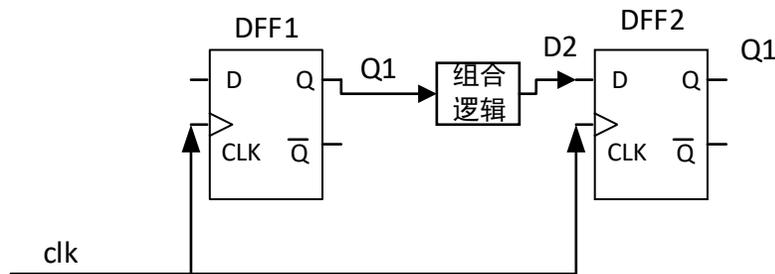
2. 用一个mux和一个反相器实现xor, 画电路or verilog代码实现



A=0时, out=b;
A=1时, out=~b



3.如下图所示：clk到Q的输出延时为 $T1_{min} < T1 < T1_{max}$ ，
 组合逻辑的延时为： $T2_{min} < T2 < T2_{max}$ ，时钟周期是 T 。满足约束条件下，求寄存器setup和holdup。



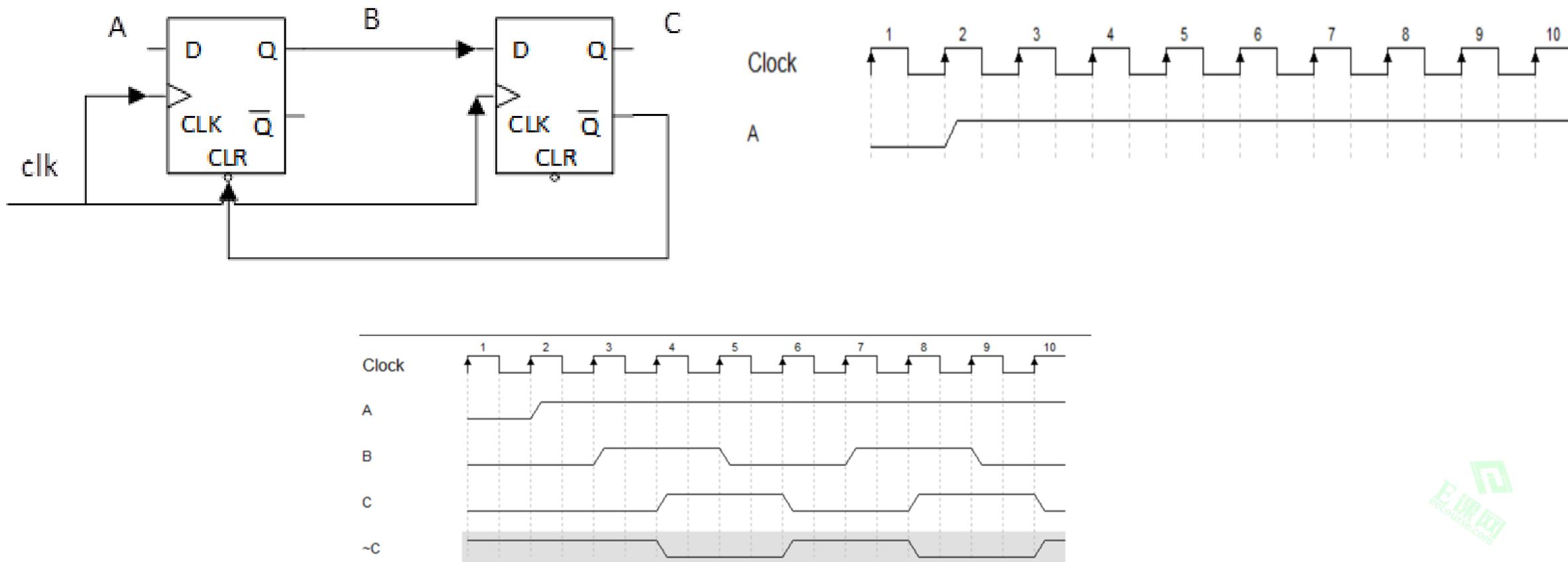
$$T1_{max} + T2_{max} + T_{setup} < T$$

故： $T_{setup} < T - T1_{max} - T2_{max}$

$$T1_{min} + T2_{min} > T_{holdup}$$



4.如下图所示：根据A的波形，画出B,C的波形，寄存器是同步清零的。



5.十进制的12.8125用二进制的数表示, 十六进制的A8D6用八进制的数表示

$12=1100$; $0.8125 \times 4=3.25$, 表示: 11.01, 故 $0.8125=0.1101$
故: $12.8125=1100.1101$

$\text{`hA8D6}=\text{`b1010100011010110}=\text{`o124326}$



6.sv比verilog强在哪?ovm uvm vmm是什么?

sv支持面向对象的编程, 支持断言的语法, 支持更多的数据类型,

vmm:verification methodology manual由ARM和Synopsys提出的, 寄存器解决方案RAL机制

ovm: open verification methodology Mentor和Candence共同提出的, 引进了factory机制

uvm: Universal Verification Methodology通用验证方法学, Accellera推出的,

继承了vmm和ovm的优点, 克服了二者的缺点, 代表了验证方法学的主流。

并且得到了三大EDA厂商Cadence、Synopsys和Mentor Graphics支持。



7.用python或者perl写程序,在xxx.log中找到fail单词

```
my %input_file = "xxx.log";  
my $inf;  
open($inf,"<",$input_file)||die "cannot open $input_file";  
while(<$inf>){  
my $input_line =$_;  
$input_line =~ m/fail/;  
print "匹配的字符:$&\n";  
}  
close $inf;
```



```
8. int b =100;  
   int *a;
```

a=&b是什么意思? a=b是什么意思?

答: a = &b 定义指针变量a, 变量指向整型数据b的地址, printf("%d",* a)结果为100;
a = b 表示指针变量a的值为100, *a表示一个地址为100的数据。



```
9. #include <stdio.h>
int main(void) {
    char **p;
    short c;
    char *m[] = {"hello world/n",
                "123456789",
                "abcdefg"};
    p = m;
    c = short(0x1000);
    printf("%c\n", **p++);
    printf("%s\n", *m);
    printf("%d\n", c);
    return 0;
}
```

```
h
hello world/n
4096
-----
Process exited after 0.1611 seconds with return value 0
请按任意键继续. . .
```



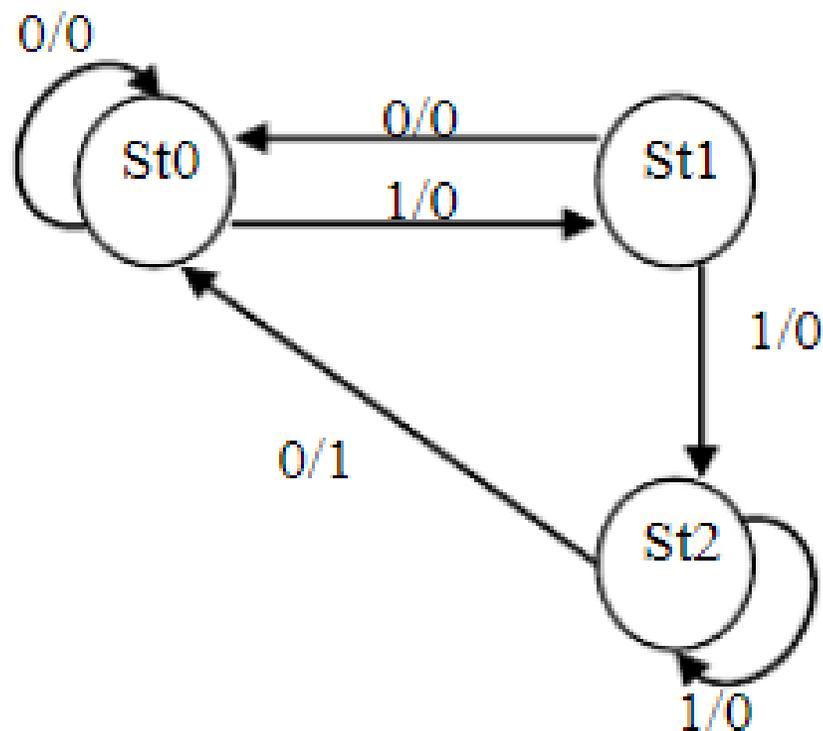
10. cpu cache miss/hit区别

CPU要访问的数据在Cache中有缓存，称为“命中” (Hit)，反之则称为“缺失” (Miss)。

当运算器需要从存储器中提取数据时，它首先在最高级的cache中寻找然后在次高级的cache中寻找。如果在cache中找到，则称为命中hit；反之，则称为不命中miss。



11. 110序列选择器 画状态图 写verilog代码



```

module sequential_detector(
input    clk    ,
input    rst_n  ,
input    in     ,
output   wire  out
);
reg [1:0] current;
reg [1:0] next;
reg out_b;
parameter s0=2'b00,s1=2'b01,s2=2'b10;

always@(posedge clk or negedge rst_n)
begin
    if(!rst_n)
        current<=s0;
    else
        current<=next;
end

```

```

always@(current or in)
begin
    case(current)
        s0:begin next = (in==1'b1)?s1:s0;end
        s1:begin next = (in==1'b1)?s2:s0;end
        s2:begin next = (in==1'b1)?s2:s0;end
    endcase
end

always@(posedge clk or negedge rst_n)
begin
    if(!rst_n)
        out_b <= 1'b0;
    else
        begin
            out_b <= 1'b0;
            case(current)
                s0:out_b <= 1'b0;
                s1:out_b <= 1'b0;
                s2:if(in==1'b1)
                    out_b <= 1'b0;
                    else
                        out_b <= 1'b1;
            endcase
        end
end
assign out = out_b;
endmodule

```



12. 一道逻辑推理题

某夜,A、B、C、D四人要过桥,该桥每次只能通行2个人,只有一个手电筒,过桥必须持有手电筒,A、B、C、D四人单独过桥的时间分别为1分钟、2分钟、5分钟、10分钟.现要求四人最短时间完成过桥,不能折返.请问该如何操作?

答: A和B过桥, A回来, C和D过桥, B回来, A和B再过桥;
用时 $2+1+10+2+2=17\text{min}$

