

参照循环语句中的例子，完成一个 32 位二进制数的乘法。

```
`timescale 1ns / 1ps
```

设计思路:

输入为两个乘数，输出为乘法结果。

由于是 32 位乘 32 位，结果应设计 64 位以免溢出。

使用阻塞赋值方式，循环 32 次（使用 i 作为循环控制变量），每次将 a 左移一位，b 右移一位，若此时 b[0] 为 1，则需要把 a 的结果累加到 out 中。

通过 `assign ans = out` 把寄存器结果赋值给输出。


```
`timescale 1ns / 1ps
```

设计思路:

初始化为 0，每 5 个时钟周期将新的随机 32 位整数赋值给 ta 和 tb，将 ta 和 tb 作为 multiplier 的输入，tans 作为输出。

观察结果。

仿真测试结果:

Name	Value	3,482 ns	3,484 ns	3,486 ns	3,488 ns	3,490 ns	3,492 ns	3,494 ns	3,496 ns	3,498 ns
> [ta:31:0]	0b636716	9743b2e		0b636716			431ee2		96ec2a2d	
>  [ta:31:0]	4w156f9c	90138420		90138420			491f895		02ba4905	
> [tans:63:0]	0668c3dbd4363ac0	2d232473264800		0668c3db4363ac0		b231c58c13da6ea		019ffdd210fff7f1		

举例分析：

3485ns 处，ta=0x0b636716，tb=0x90138420，tsum=0x0668c83db4363ac0。

通过运行 16 进制计算器，发现答案正确，以下是计算过程。

0b636716 = 191063830(Base 10)

90138420 = 2417198112(Base 10)

$191063830 \times 2417198112 = 461839129147488960$ (Base 10) = **668c83db4363ac0(Hex)**

3490ns 处，ta=0xe131eec2，tb=0xca91f895，tsum=0xb231ce58c13de6ea。

通过运行 16 进制计算器，发现答案同样正确，以下是计算过程。

e131eec2 = 3778145986(Base 10)

ca91f895 = 3398563989(Base 10)

$3778145986 \times 3398563989 = 12840270893204498154$ (Base 10) = **b231ce58c13de6ea(Hex)**

因此，验证成功！