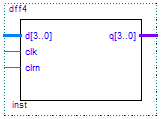
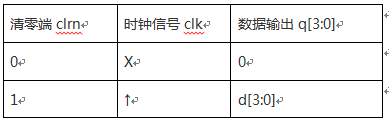
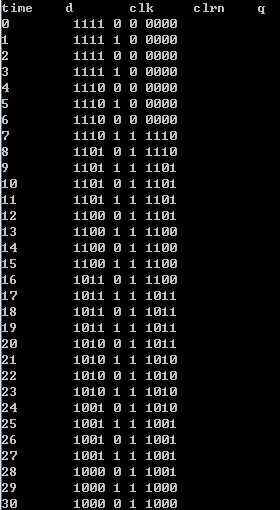
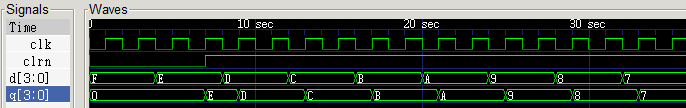
* **任务描述**
* 本关需要你根据所学的触发器及时序电路的知识完成4位寄存器的设计。熟悉Verilog HDL文本设计流程，根据寄存器逻辑真值表，完善4位寄存器电路的功能描述风格Verilog HDL 代码。
* **相关知识**
* **逻辑原理**
* 4位寄存器能够存储4位二进制代码，受到一个时钟信号clk控制，当clk的触发沿到来时，输出信号等于输入信号。。
* **4位寄存器逻辑图**
* 
* clrn:为清零端，低电平有效;
* clk：时钟脉冲输入，上升沿有效；
* d[3..0]：4 位数据输入端；
* q[3..0]：4 位数据输出端；
* **4位寄存器真值表**
* 
* **编程要求**
* 根据提示，在右侧编辑器补充代码，设计一个具有异步清零功能的4位锁存器。该4位寄存器的数据输入端为d，clk为时钟端，上升沿触发；输出端为q。 当清零端clrn为0时，输出端q为0； 当清零端clrn为1，时钟信号上升沿到来时，输出信号等于输入信号。 。
* **测试说明**
* 以下是测试样例： 仿真波形如下图所示。 
* 验货啦，验货啦，开始你的任务吧！

#### 任务描述

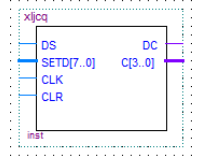
本关需要你根据所学的时序逻辑及数字电路的知识完成序列检测器的设计，验证序列检测器的逻辑功能，根据逻辑真值表和逻辑表达式完成序列检测功能。熟悉Verilog HDL文本设计流程，掌握时序逻辑电路的设计方法，最后完善序列检测器电路的功能描述风格Verilog HDL 代码。

#### 相关知识

##### 逻辑原理

序列检测器是一种同步时序电路，它用于搜索、检测输入的二进制代码串中是否出现指定的代码序列。当序列检测器连续收到一组串行二进制码后，如果这组码与检测器中预先设置的码相同,则输出为1，否则输出为0。 由于这种检测的关键在于正确码的收到必须是连续的，这就要求检测器必须记住前一次的正确码及正确序列，直到在连续的检测中所收到的每一位码都与预置数的对应码相同。在检测过程中，任何一位不相等都将回到初始状态重新检测。 例如，若预先设置的二进制码为11010110，则当序列检测器连续接收到的8位序列信号为11010110（时间顺序为自左向右）时，会输出1，表示检测到预置数据；否则输出0，表示没有检测到。

##### 序列检测器逻辑电路图

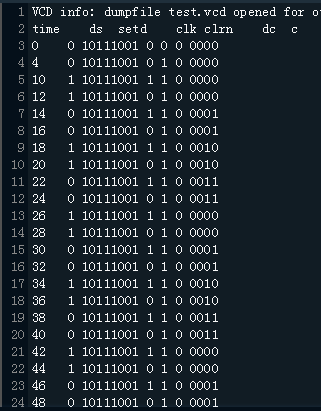
通过 Verilog HDL 语言编程，实现一个8位序列检测器， 要求有1 个串行数据输入端、1个时钟脉冲输入、1个异步清零端、1个8位预置码输入端、4位的状态码输出端和检测结果输出端，具体接口说明如下图所示。 CLR：异步清零信号，低电平有效； CLK： 序列检测器时钟脉冲输入，上升沿有效。 DS： 串行数据输入端 SETD[7..0]：8 位预置码输入端 C[3..0]： 状态码输出端 DC： 检测结果输出

#### 编程要求

为了完成序列检测任务，完善编程模块设计代码，你编写的程序要能根据不同的输入能够得到满足序列检测器真值表的逻辑输出。

#### 测试说明

以下是测试样例：



寄存器设计

* [任务描述](https://www.educoder.net/tasks/GYC2SLQL/943456/avxrukb35fs6?coursesId=GYC2SLQL" \l "%E4%BB%BB%E5%8A%A1%E6%8F%8F%E8%BF%B0)
* [相关知识](https://www.educoder.net/tasks/GYC2SLQL/943456/avxrukb35fs6?coursesId=GYC2SLQL" \l "%E7%9B%B8%E5%85%B3%E7%9F%A5%E8%AF%86)
  + [寄存器的功能](https://www.educoder.net/tasks/GYC2SLQL/943456/avxrukb35fs6?coursesId=GYC2SLQL" \l "%E5%AF%84%E5%AD%98%E5%99%A8%E7%9A%84%E5%8A%9F%E8%83%BD)
  + [寄存器的结构图](https://www.educoder.net/tasks/GYC2SLQL/943456/avxrukb35fs6?coursesId=GYC2SLQL" \l "%E5%AF%84%E5%AD%98%E5%99%A8%E7%9A%84%E7%BB%93%E6%9E%84%E5%9B%BE)
  + [如何用verilog HDL描述出寄存器的功能](https://www.educoder.net/tasks/GYC2SLQL/943456/avxrukb35fs6?coursesId=GYC2SLQL" \l "%E5%A6%82%E4%BD%95%E7%94%A8verilog hdl%E6%8F%8F%E8%BF%B0%E5%87%BA%E5%AF%84%E5%AD%98%E5%99%A8%E7%9A%84%E5%8A%9F%E8%83%BD)
* [编程要求](https://www.educoder.net/tasks/GYC2SLQL/943456/avxrukb35fs6?coursesId=GYC2SLQL" \l "%E7%BC%96%E7%A8%8B%E8%A6%81%E6%B1%82)
* [测试说明](https://www.educoder.net/tasks/GYC2SLQL/943456/avxrukb35fs6?coursesId=GYC2SLQL" \l "%E6%B5%8B%E8%AF%95%E8%AF%B4%E6%98%8E)

### **任务描述**

本关任务：编写一个异步清零带写使能端的32位寄存器。

### **相关知识**

CPU内的隐含寄存器是一些特殊的寄存器, 对用户是不可用的,专供控制器使用。 在指令执行过程中，由硬件自动控制这些寄存器。包括程序计数器 PC 、指令寄存器 IR 、程序状态字 PSW 、存储器地址寄存器 MAR 、存储器数据寄存器 MDR 等。其中： 指令寄存器 IR :用来存放当前正在执行的指令代码。 程序计数器 PC :也称为指令指针,用来指示将要执行 的下一条指令在内存中的地址。

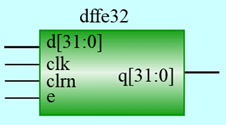
为了完成本关任务，你需要掌握：1.寄存器的功能，2.如何用verilog HDL描述出寄存器的功能。

#### **寄存器的功能**

寄存器是计算机中实现数据存放部件，主要完成数据的临时存储。

异步清零带写使能端的32位寄存器能实现异步（不受时钟的控制）清零功能。当写使能输入端e=1，在时钟信号的上升沿到来时，将32位的输入数据d写入寄存器。 32位的数据输入端d，用于写入数据； e为写使能输入端，e=1执行写操作； clk为时钟信号输入端，上升沿触发； clrn为异步清零信号输入端，clrn=0时清零。

#### **寄存器的结构图**



#### **如何用verilog HDL描述出寄存器的功能**

（1）always用来描述一些比较复杂的组合逻辑和时序逻辑电路模块 always语句的格式如下：

1. always @(<敏感信号表达式>)
2. begin
3. // 过程赋值语句
4. // if语句
5. // case语句
6. end

@后跟的是表示在什么情况下触发执行。若敏感信号表达式为“\*”，则表示所有输入信号有变化时都触发。触发可以是电平触发，也可以是上升沿或者是下降沿触发，分别跟posedge和negedge，如

1. always @(posedge clk0 or negedge clk1)

（2）if-else语句 if-else语句判定所给条件（表达式）是否满足，根据判定的结果（真或假）决定执行给出的两个操作之一。若表达式的值为0或z，则判定的结果为“假”；若为1，则结果为“真”。if-else语句的格式如下：

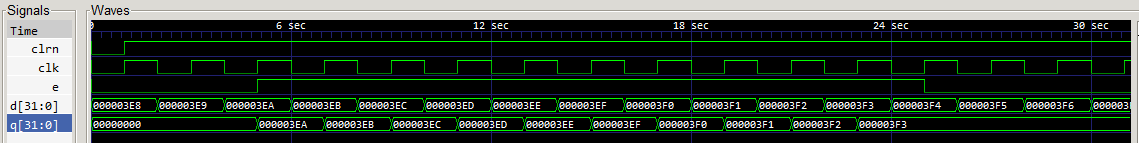
1. if (表达式)
2. 语句1; //表达式的值=1
3. [else //表达式的值=0 or z
4. 语句2;]

其中，语句1、语句2可以是任何一个语句，若是多个语句，需要用begin-end将这些语句括起来；else和语句2可以没有。

### **编程要求**

根据提示，在右侧编辑器补充代码，设计一个异步清零带写使能端的32位寄存器。

### **测试说明**

平台会对你编写的代码进行测试。 仿真波形如下： 

开始你的任务吧，祝你成功！