

## 平台

软件: ModelSim-Altera 6.5e (Quartus II 10.0) Starter Edition

## 内容

### 1 设计流程

使用 ModelSim 仿真的基本流程为:

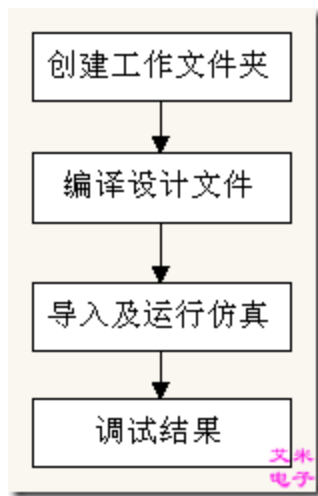


图 1.1 使用 ModelSim 仿真的基本流程

## 2 开始

### 2.1 新建工程

打开 ModelSim 后，其画面如图 2.1 所示。

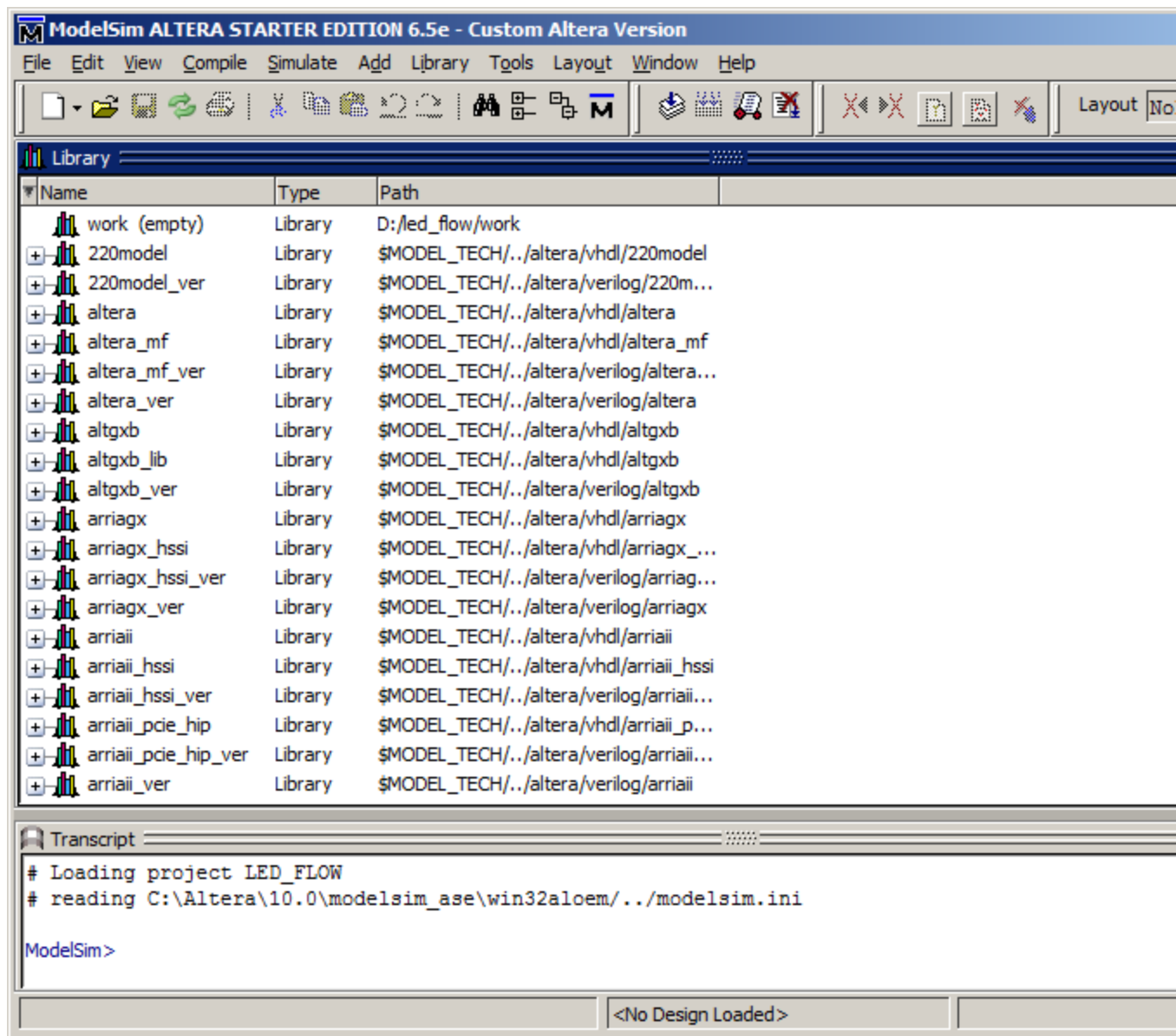


图 2.1 ModelSim 画面

1. 选择 **File>New>Project** 创建一个新工程。打开的 **Create Project** 对话框窗口，可以指定工程的名称、路径和缺省库名称。一般情况下，设定 **Default Library Name** 为 **work**。指定的名称用于创建一个位于工程文件夹内的工作库子文件夹。该对话框如图 2.2 所示，此外还允许通过选择.ini 文件来映射库设置，或者将其直接拷贝至工程中。

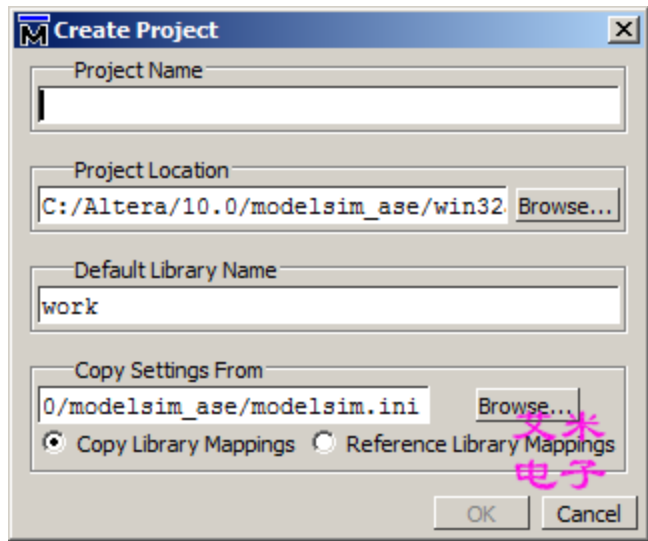


图 2.2 创建工程的对话框

2. 按照图 2.3 所示，设置 **Project Name** 为 *LED\_FLOW*，**Project Location** 为 *D:/led\_flow*。

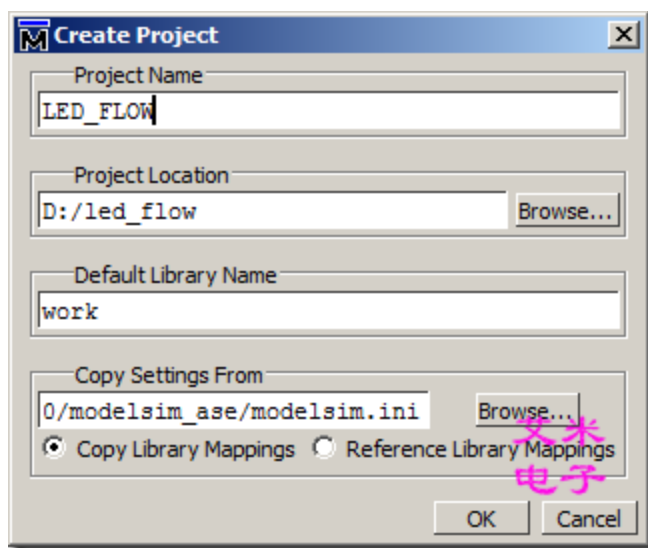


图 2.3 输入工程信息

当单击 **OK** 按钮后，在主体窗口的下方将出现 **Project** 标签，如图 2.4 所示。

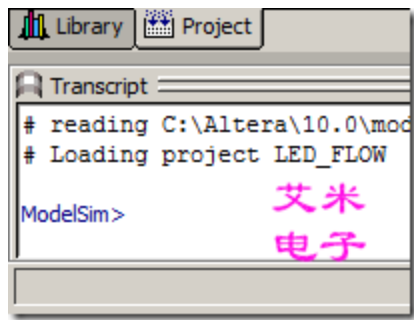


图 2.4 Project 标签

3. 之后，将出现 **Add Items to the Project** 的对话框，如图 2.5 所示。

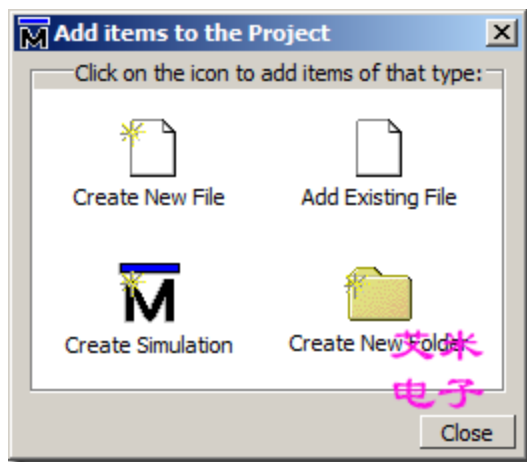


图 2.5 在工程中，添加新项目

## 2.2 在工程中，添加新项目

在 **Add Items to the Project** 对话框中，包括以下选项：

- **Create New File**——使用源文件编辑器创建一个新的 Verilog、VHDL、TCL 或文本文件
- **Add Existing File**——添加一个已存在的文件
- **Create Simulation**——创建指定源文件和仿真选项的仿真配置
- **Create New Folder**——创建一个新的组织文件夹

1. 单击 **Create New File**。打开图 2.6 所示窗口。

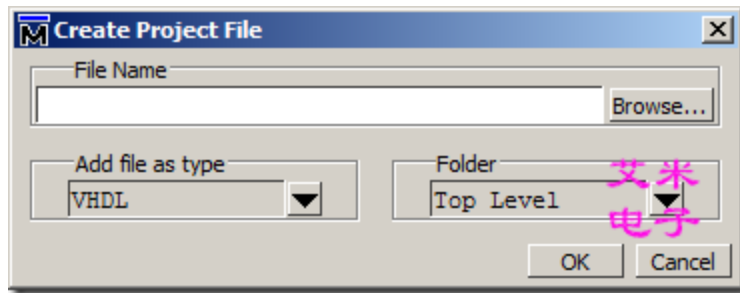


图 2.6 创建工程文件夹

2. 输入文件名称：LED\_FLOW，然后选择文件类型为 Verilog。

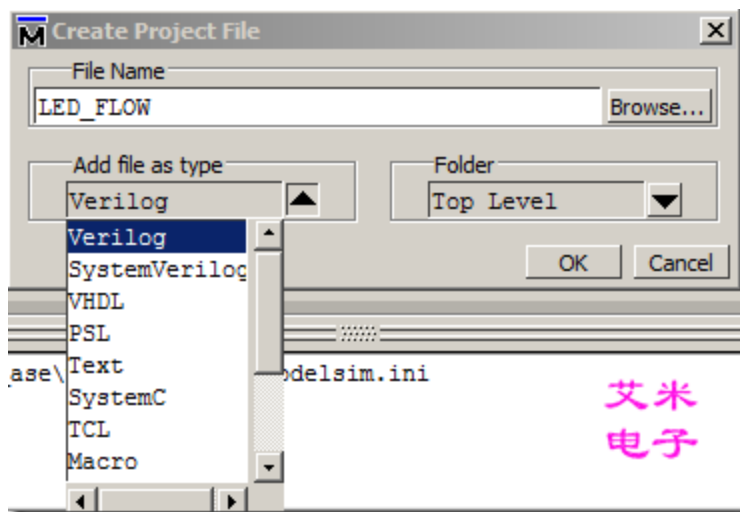


图 2.7 输入工程文件信息

3. 单击 **OK**，关闭本对话框。新的工程文件将会在工程窗口显示。单击 **Close**，以关闭 **Add Items to the Project**。

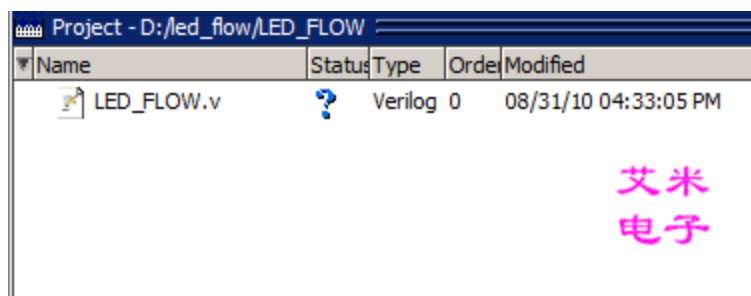


图 2.8 新的设计文件 LED\_FLOW.v

4. 双击打开 LED\_FLOW.v 文件（注意：若是 Verilog 文件已经关联了其他的文本编辑器，则双击后在关联的文本编辑器中打开）。

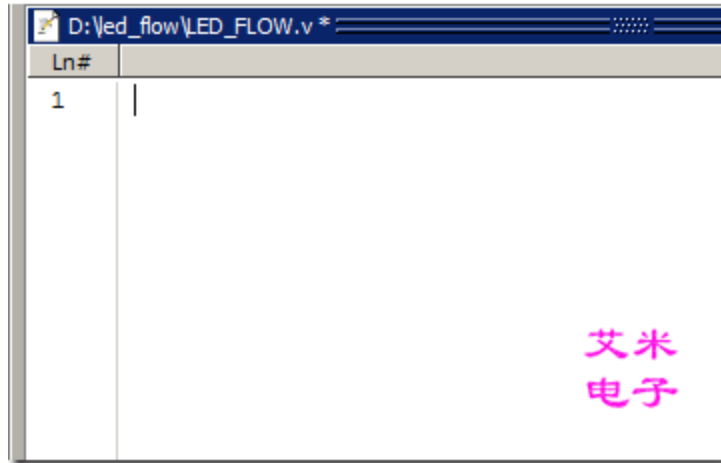


图 2.9 LED\_FLOW 代码输入窗口

在 LED\_FLOW.v 输入下面的测试平台代码：

```
`timescale 1ns/1ns
module LED_FLOW;
reg CLOCK_50M;
reg RST_N;
wire [9:0] LED;
led led_inst
(
    .clk_50M(CLOCK_50M),
    .reset_n(RST_n),
    .led(LED)
);

initial
begin
    CLOCK_50M = 0;
    while (1)
        #10 CLOCK_50M = ~CLOCK_50M;
end

initial
begin
    RST_N = 0;
    while (1)
```

```

        #10 RST_N = 1;
    end

    initial
    begin
        $display($time,"CLOCK_50M=%d RST_N=%d LED =%d", CLOCK_50M,
RST_N, LED);
    end

endmodule

```

录入完代码后，单击 **Save**。

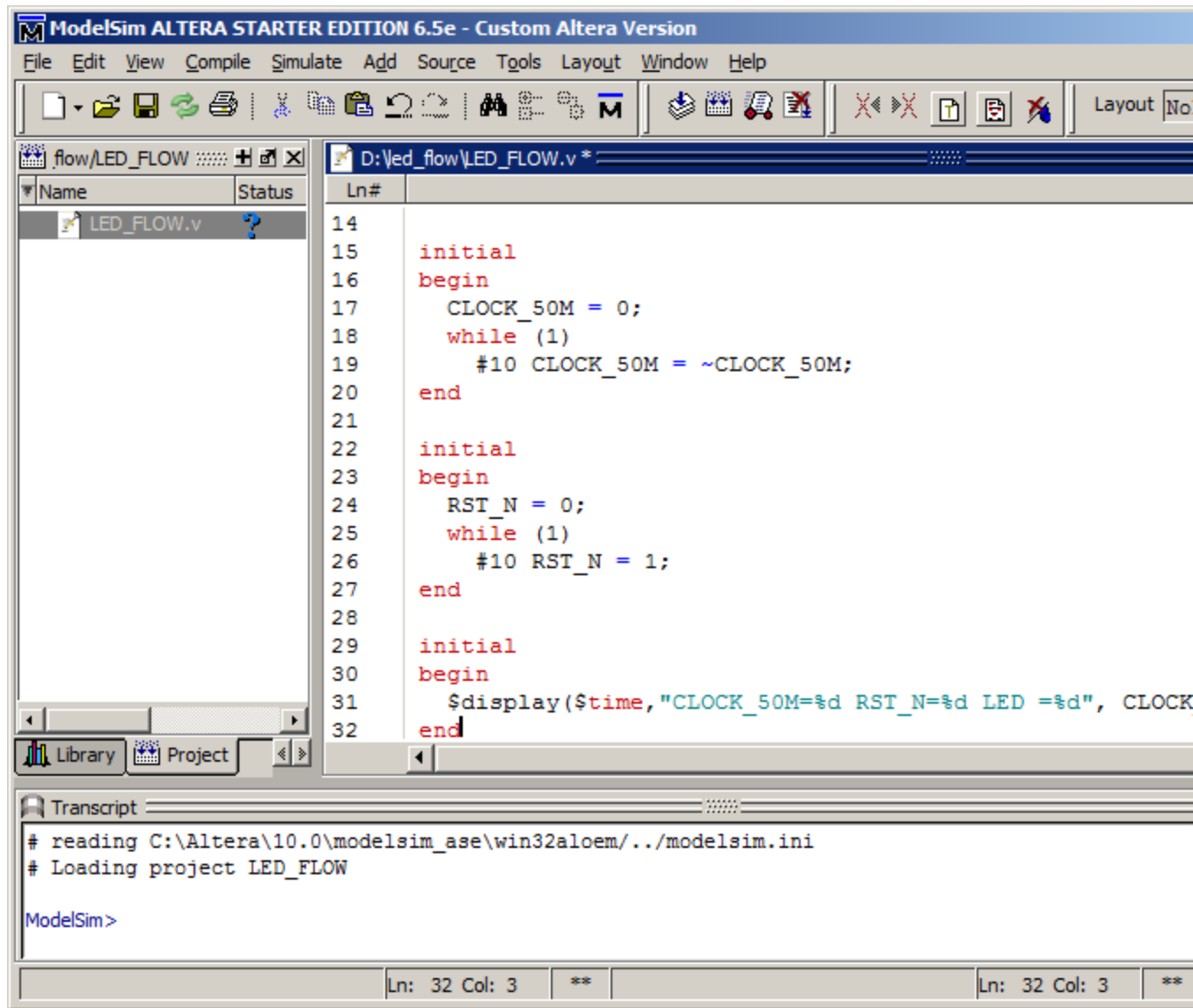


图 2.10 输入 testbench 代码

5. 选择 **File>New>Source>Verilog**，创建新的 Verilog 文件，如图 2.11 所示。

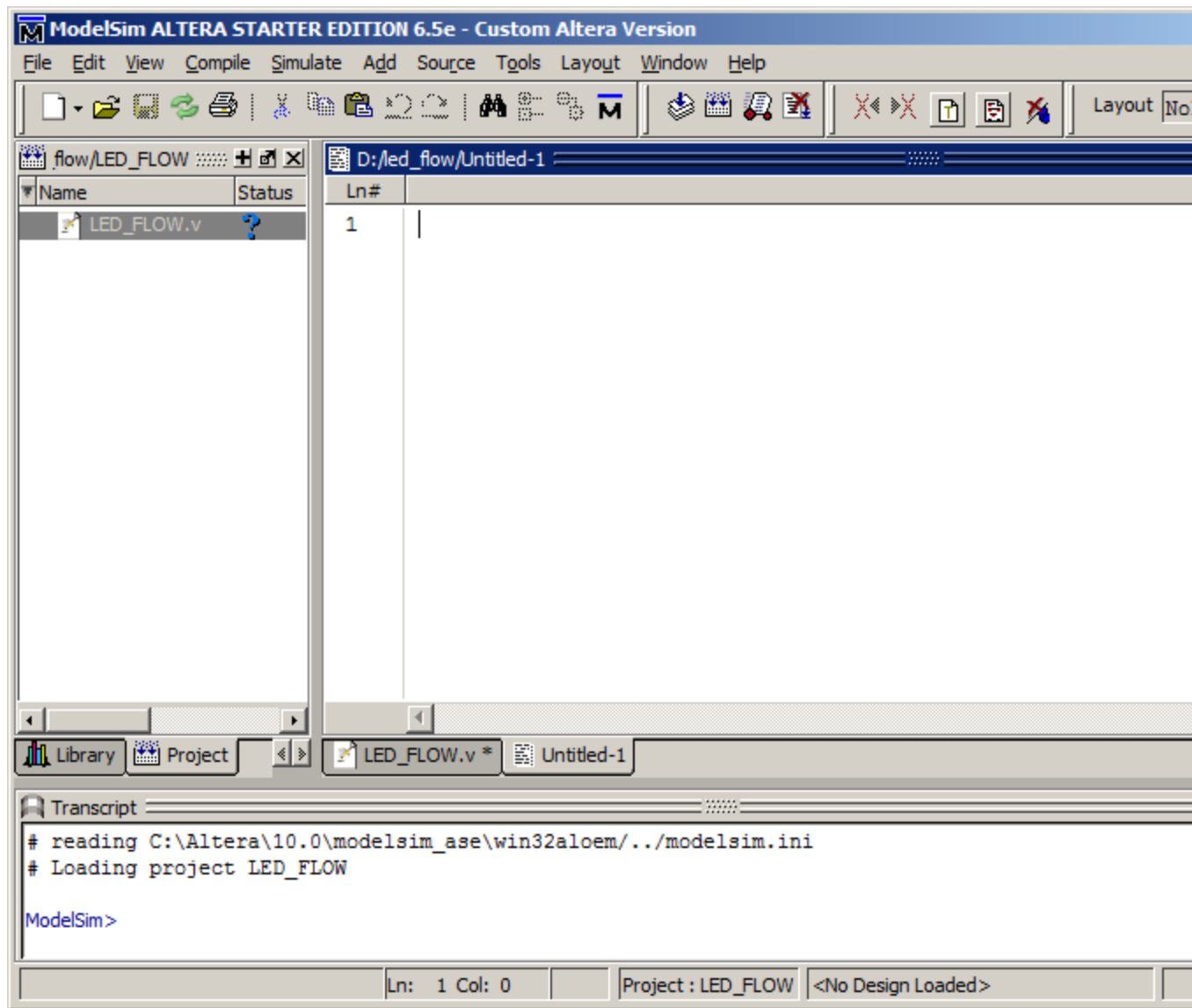


图 2.11 创建新的 Verilog 文件

6. 录入下面的代码，录入画面如图 2.12 所示。

```
`timescale 1ns/1ns
module led(
    input clk_50M,           // System clock 50MHz
    input reset_n,           // System reset
    output reg [9:0] led      // led
);
```



```

reg [13:0] counter = 0;
reg [3:0] state = 0;
always @ (posedge clk_50M, negedge reset_n)
    if (!reset_n)
        counter <= 0;
    else
        counter <= counter + 1'b1;
always @ (posedge counter[13])
    if (!reset_n)
        state <= 0;
    else
        begin
            if (state == 4'b1001)
                state <= 0;
            else
                state <= state + 1'b1;
        end
always @ (posedge clk_50M, negedge reset_n)
    if (!reset_n)
        led <= 0;
else
    begin
        case (state)
            4'b0000: led <= 10'b00000_00001;
            4'b0001: led <= 10'b00000_00010;
            4'b0010: led <= 10'b00000_00100;
            4'b0011: led <= 10'b00000_01000;
            4'b0100: led <= 10'b00000_10000;
            4'b0101: led <= 10'b00001_00000;
            4'b0110: led <= 10'b00010_00000;
            4'b0111: led <= 10'b00100_00000;
            4'b1000: led <= 10'b01000_00000;
            4'b1001: led <= 10'b10000_00000;
            default: led <= 10'b00000_00001;
        endcase
    end
end

```

endmodule

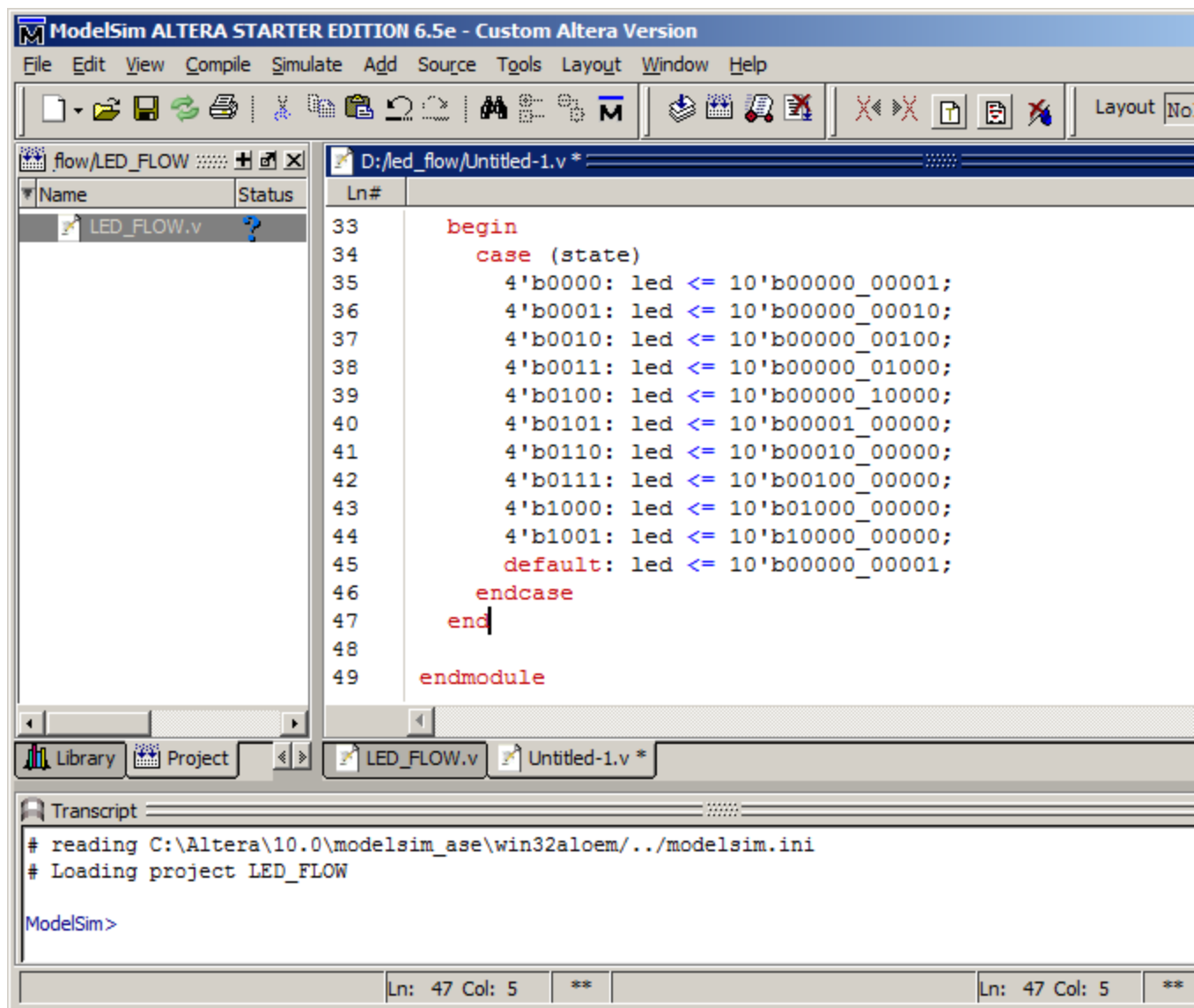


图 2.12 录入新文件

7. 选择 **File>Save**, 输入文件名: `led.v`, 单击 **Save**, 如图 2.13 所示。

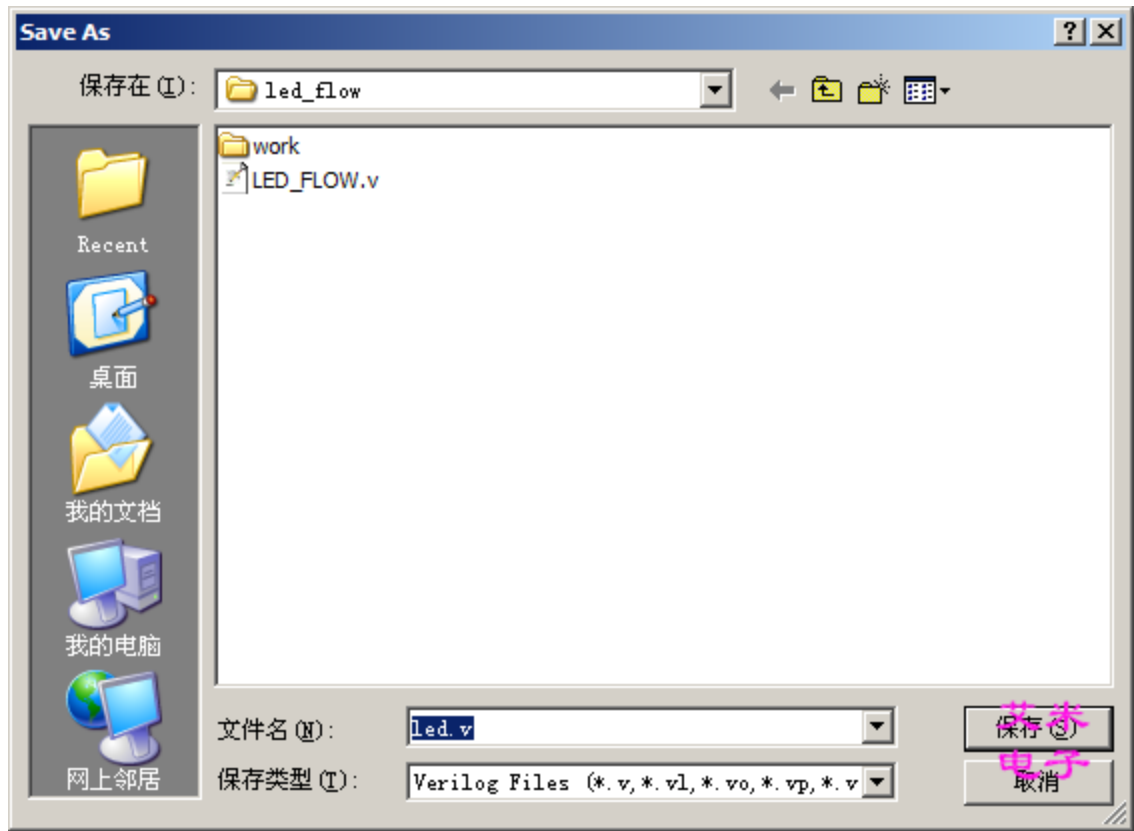


图 2.13 保存 led.v

8. 选择 **Project>Add to Project>Existing File**，如图 2.14 所示。

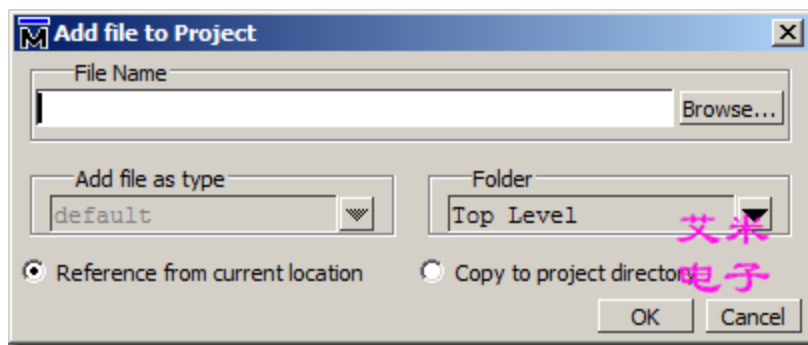


图 2.14 添加文件到工程中

9. 单击 **Browse**，选择 led.v，如图 2.15 所示。

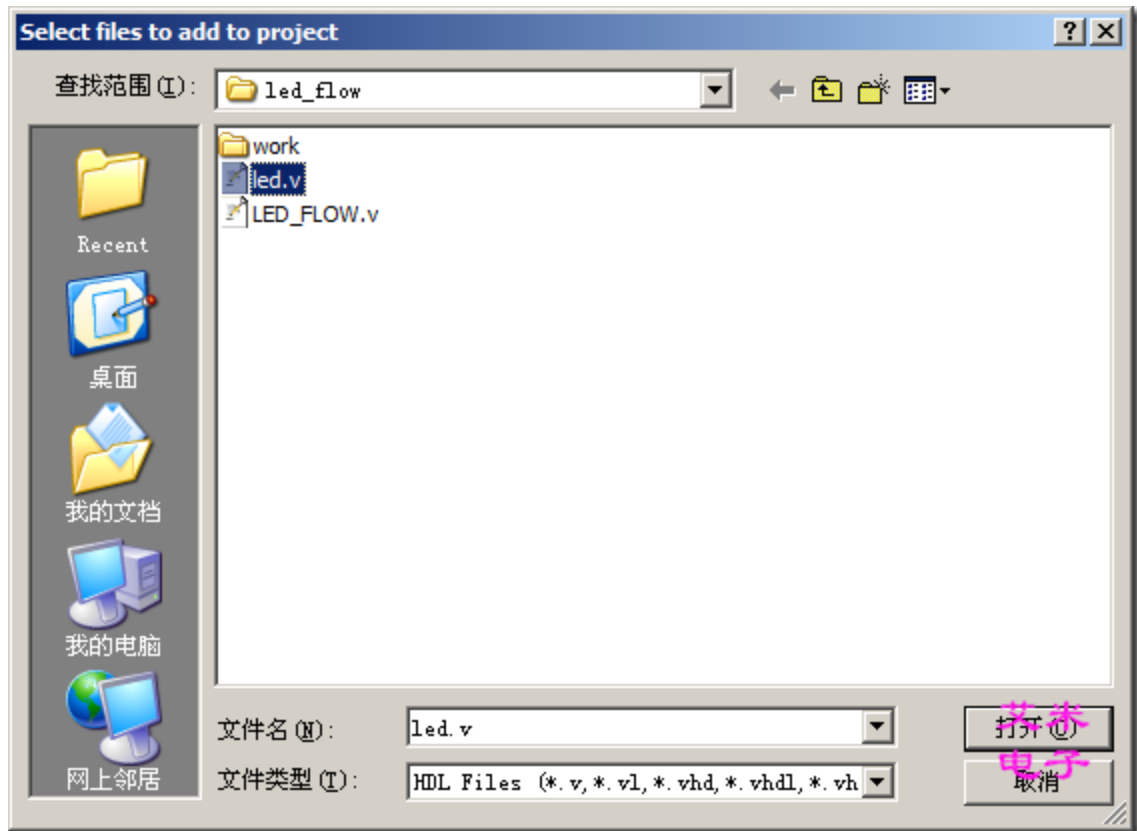


图 2.15 选择待加入工程的文件

10. 单击打开，在 **Add file to the project** 窗口，单击 **OK**。

### 2.3 编译文件

在 **Project** 标签下的 **Status** 列的问号，表示文件尚未编译进工程，或者在最后编译前，源文件有所改动。欲编译文件，选择 **Compile<Compile ALL**，或者右击 **Project** 标签，选择 **Compile>Compile All**。

1. 倘若此处没有错误，编译成功的消息，就会在 **Transcript** 窗口如图 2.6 所示。

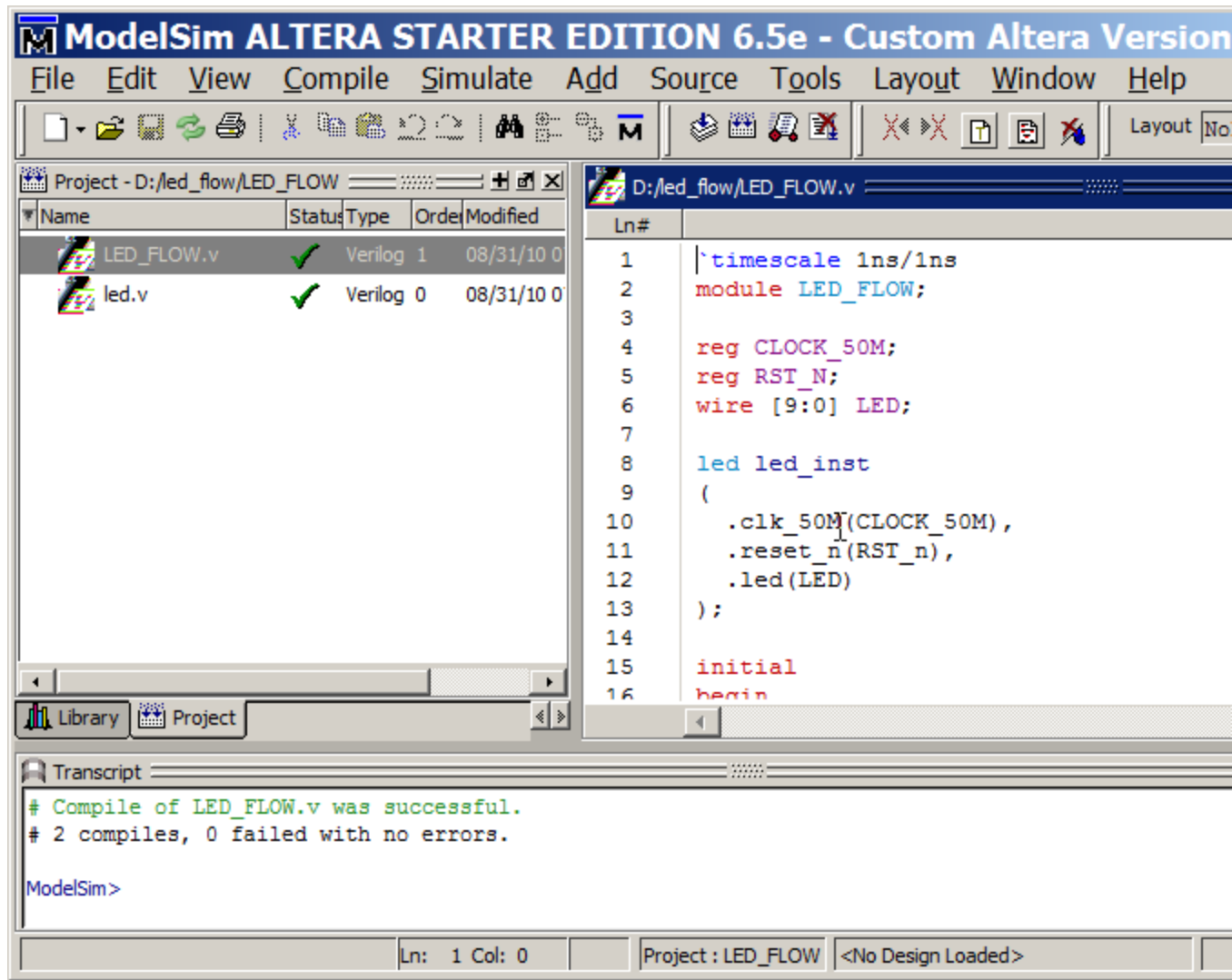


图 2.16 编译成功

### 3 仿真工程

#### 3.1 开始仿真

1. 单击 **Library** 图标，选择 **work**，单击+以展开选项，然后选择 **LED\_FLOW**。单击右键，选择编译，如图 3.1 所示。

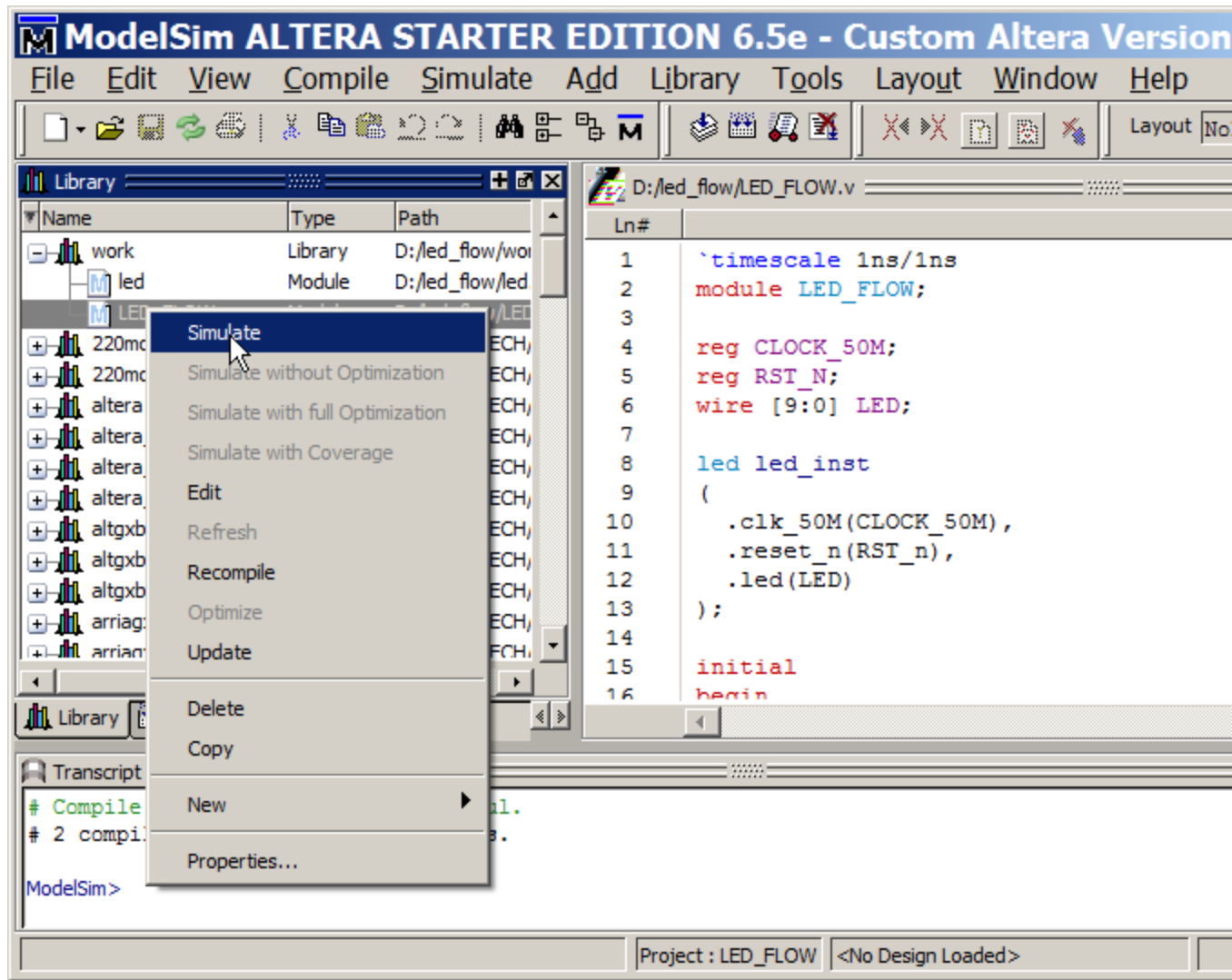


图 3.1 单击 Simulate

2. 单击 **Simulate**，到达图 3.2 所示画面。

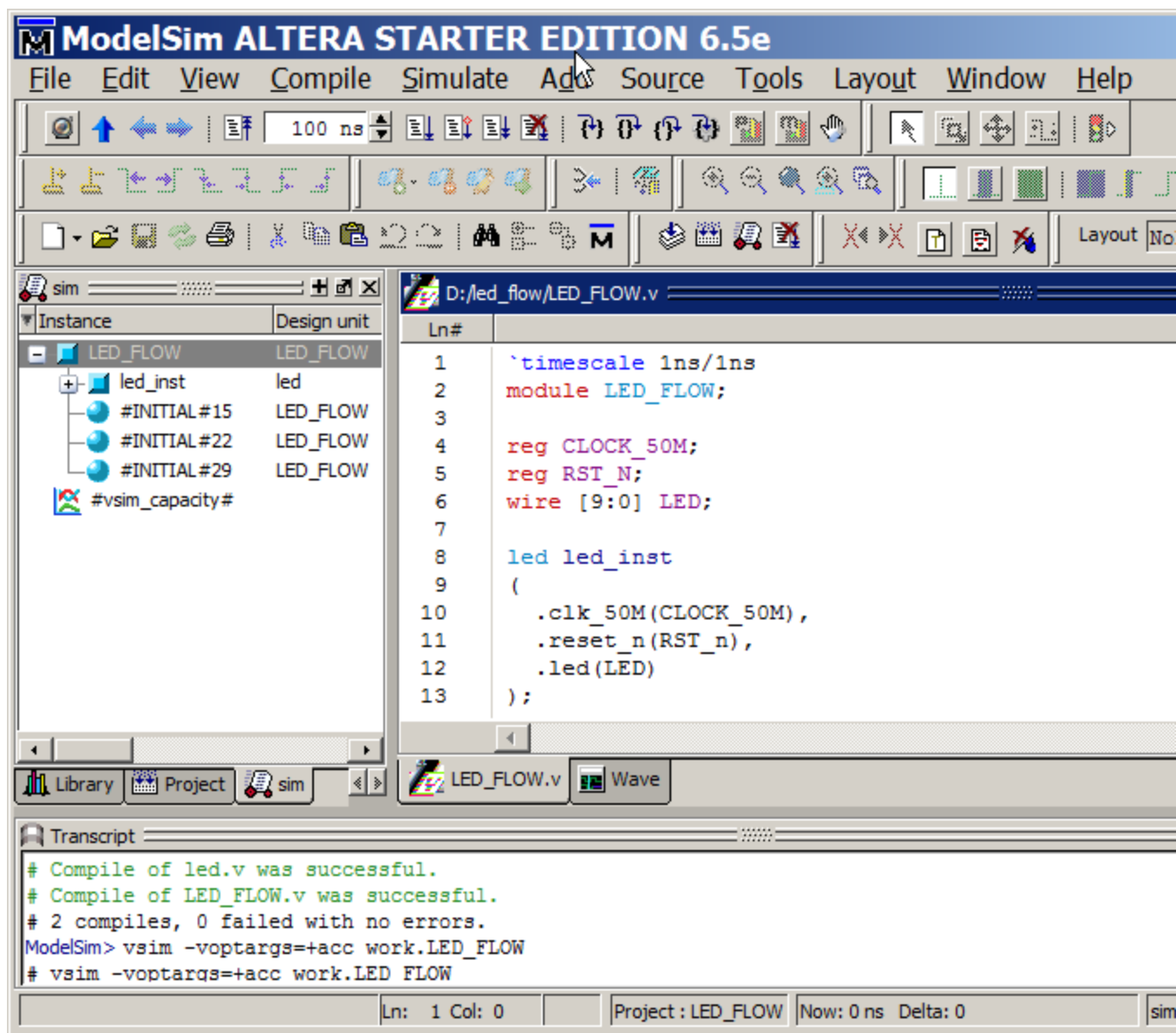


图 3.2 仿真窗口

4. 在图 3.2 中，单击 **LED\_FLOW**，单击右键，然后选择 **Add>To Wave>All Items in region**，然后单击左键。出现图 3.3 所示画面。

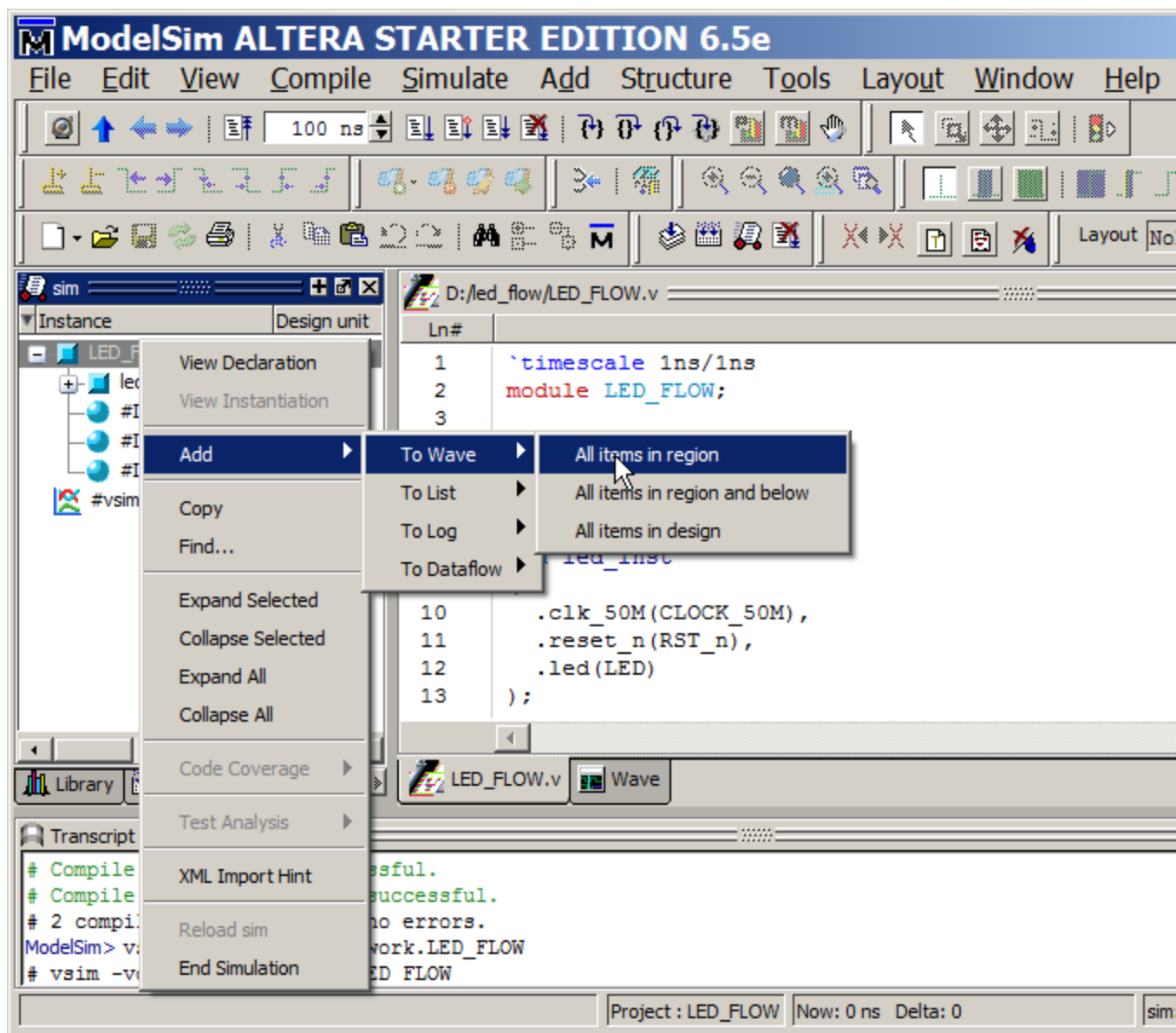


图 3.3 Add To Wave

### 3.2 仿真设置

1. 完成上述最后一步后，波形窗口出现。



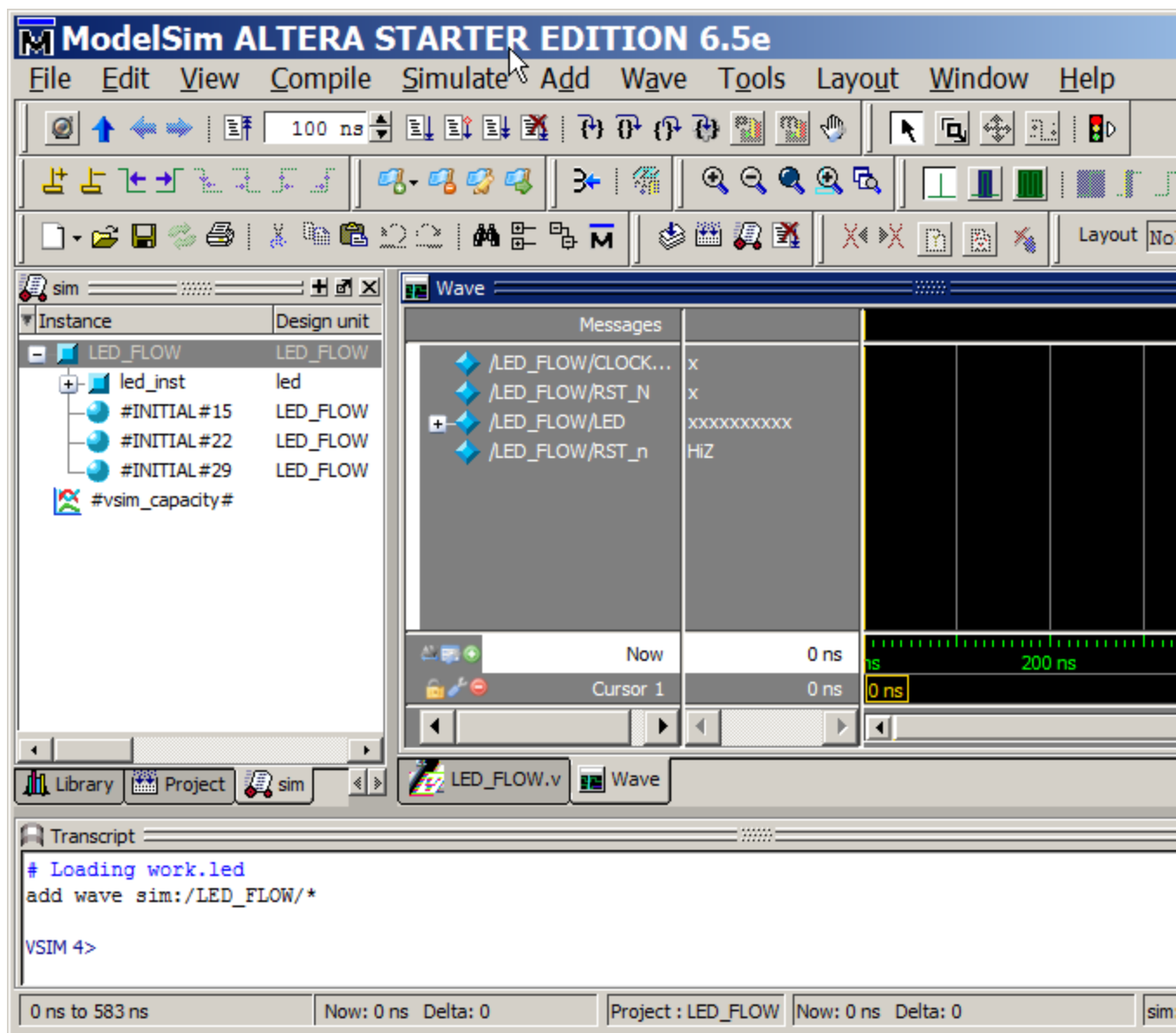


图 3.4 波形窗口

2. 在 **Run Length** 列输入仿真时间长度为 **10ms**，如图 3.5 所示。

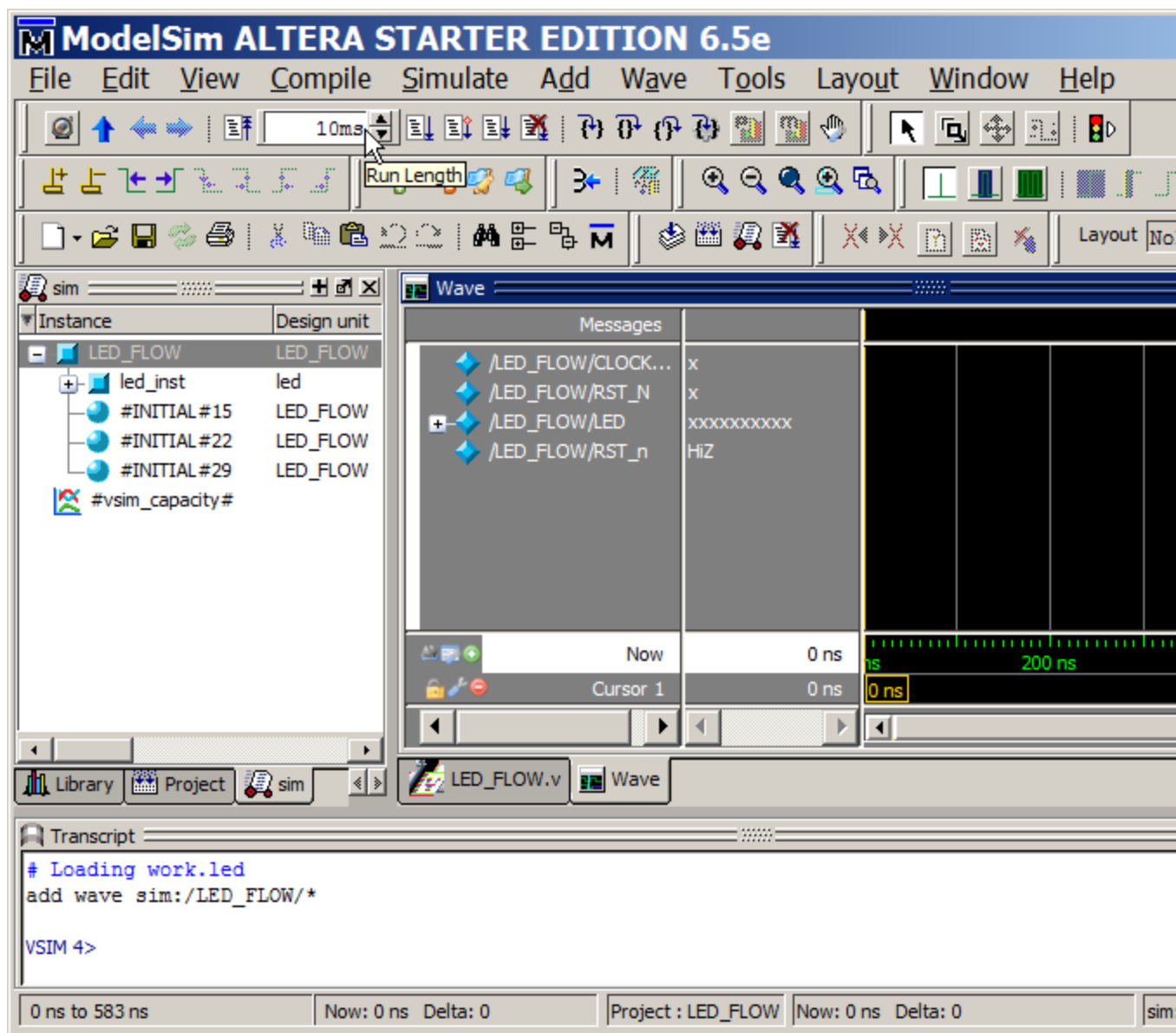


图 3.5 设置 Run Length

3. 单击 Run 按钮，如图 3.6 所示。

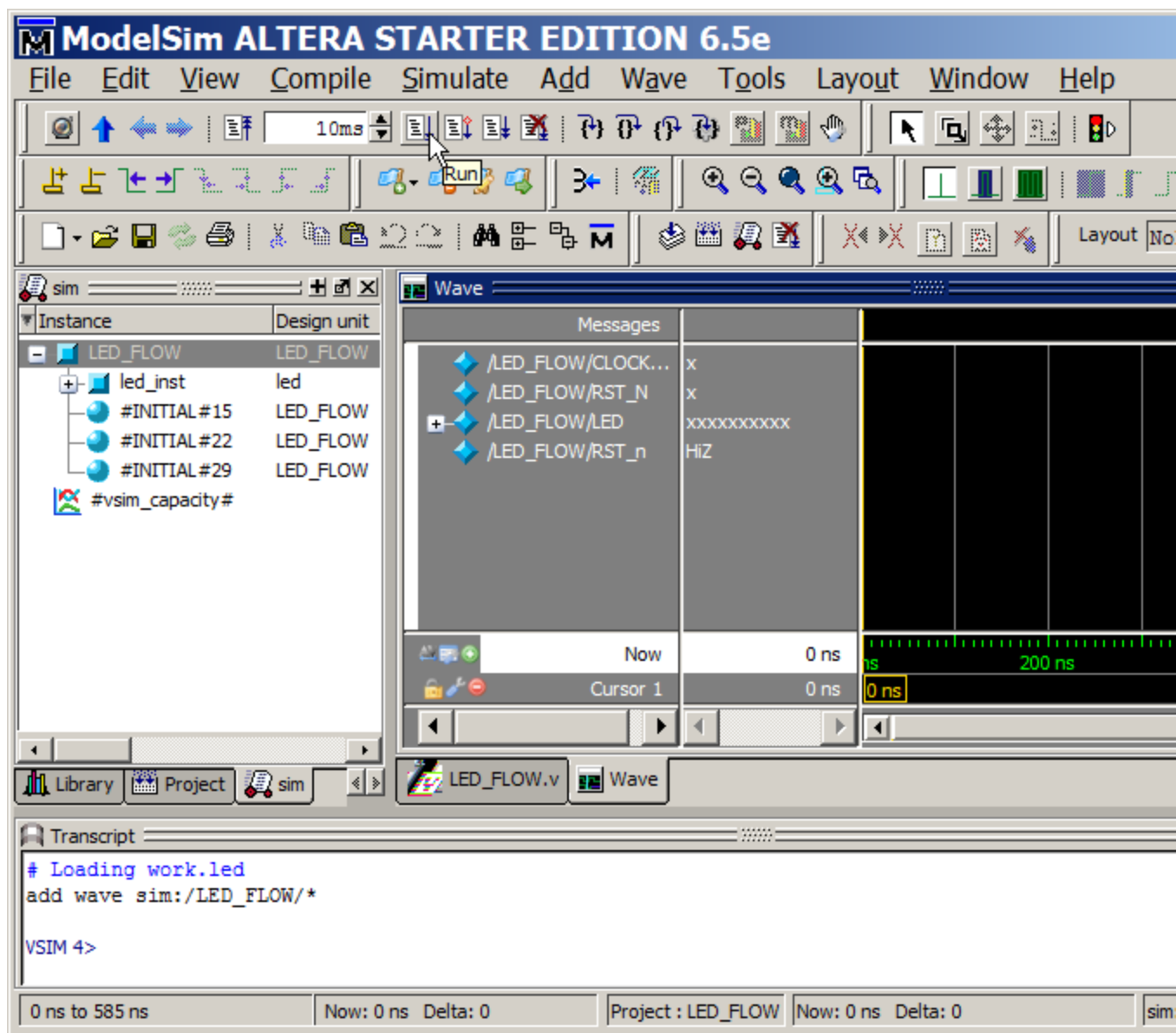


图 3.6 运行仿真

4. 运行若干秒后，将会如图 3.9 所示的仿真结果。

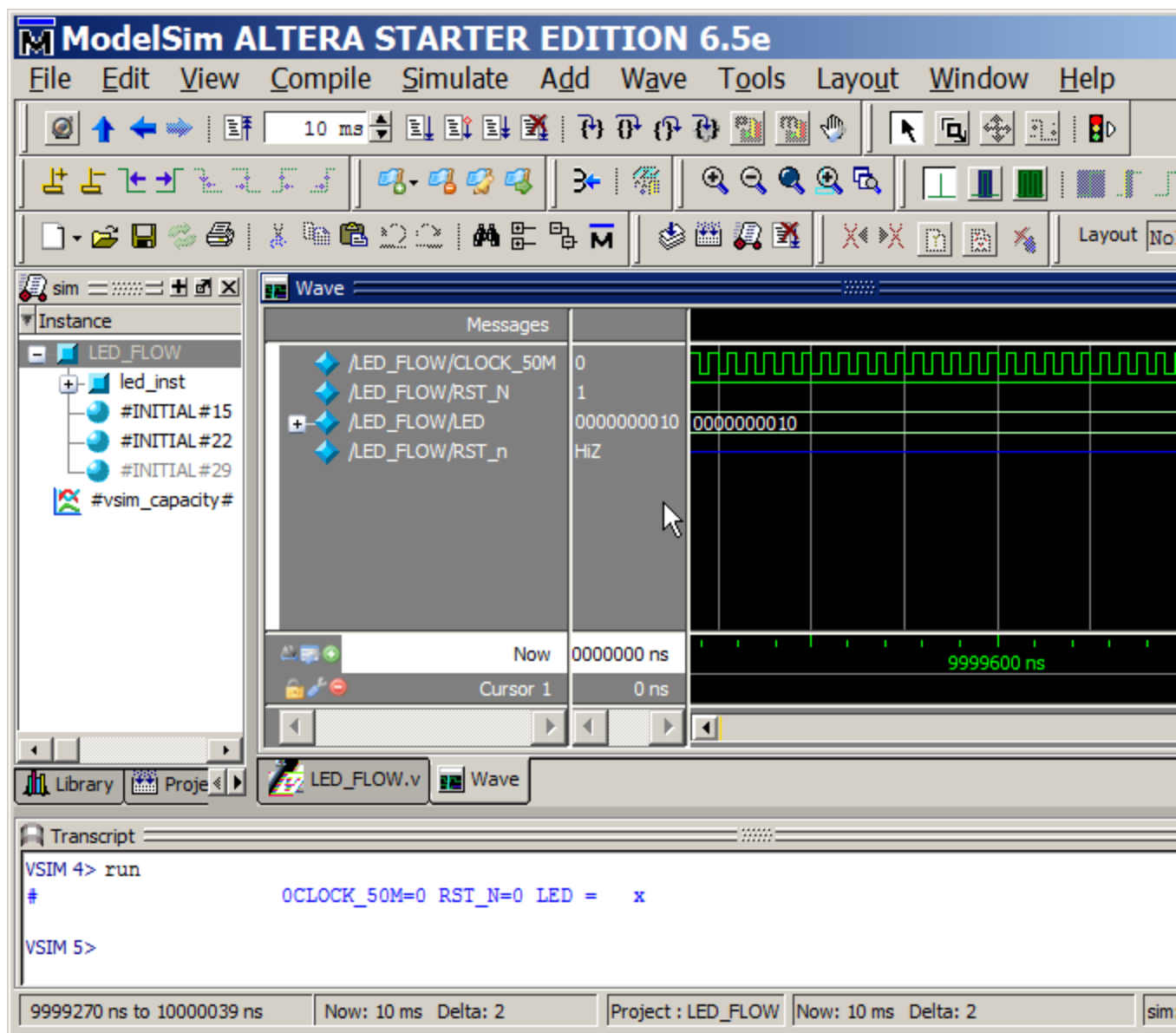


图 3.7 显示仿真结果

5. 连续单击 **Zoom Out** 图标，可查看仿真的完整波形，如图 3.8 所示。

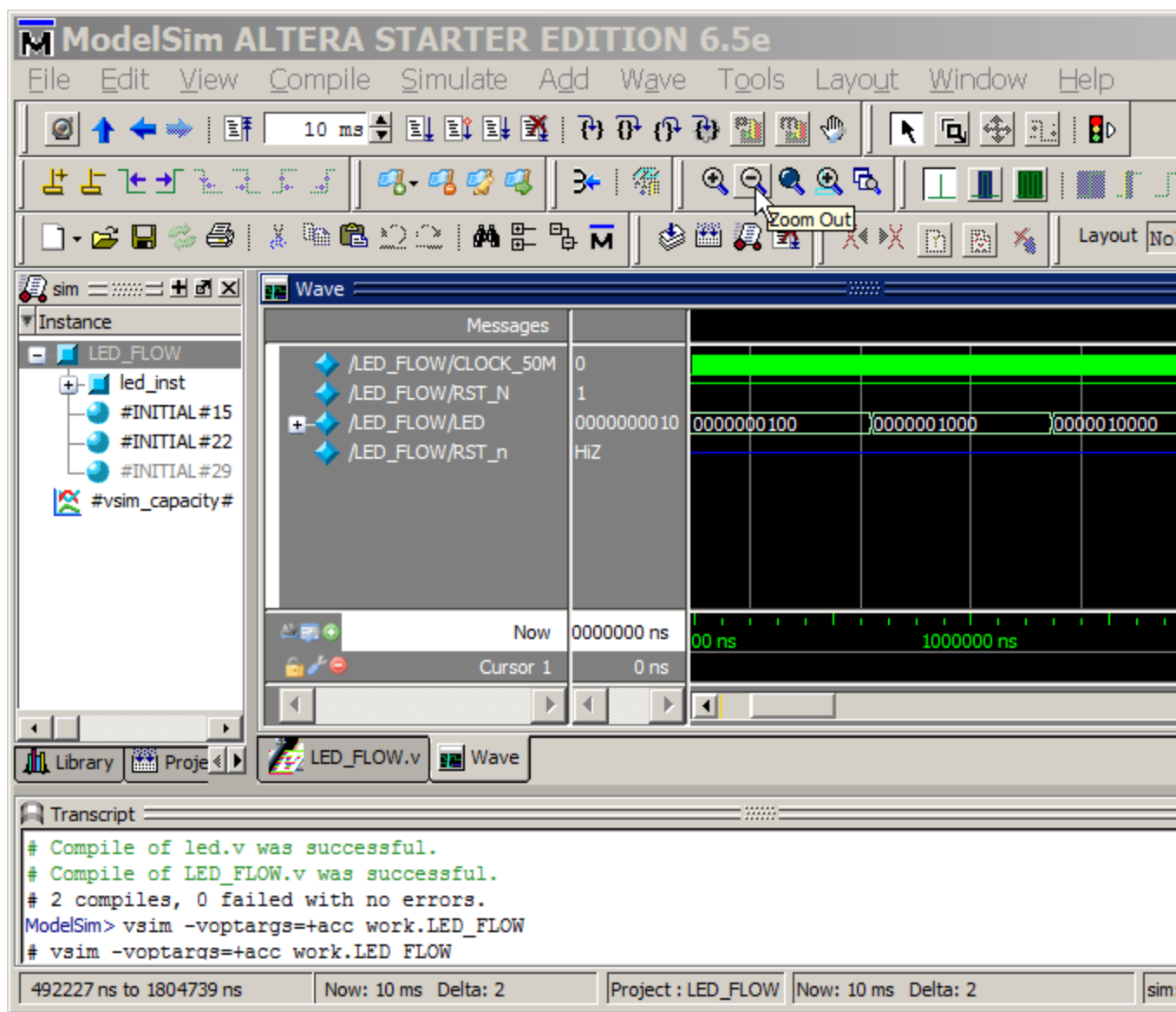


图 3.8 波形窗口

6. 单击鼠标所指的 $+$ ，展开 **LED\_FLOW** 的波形，如图 3.9 所示。

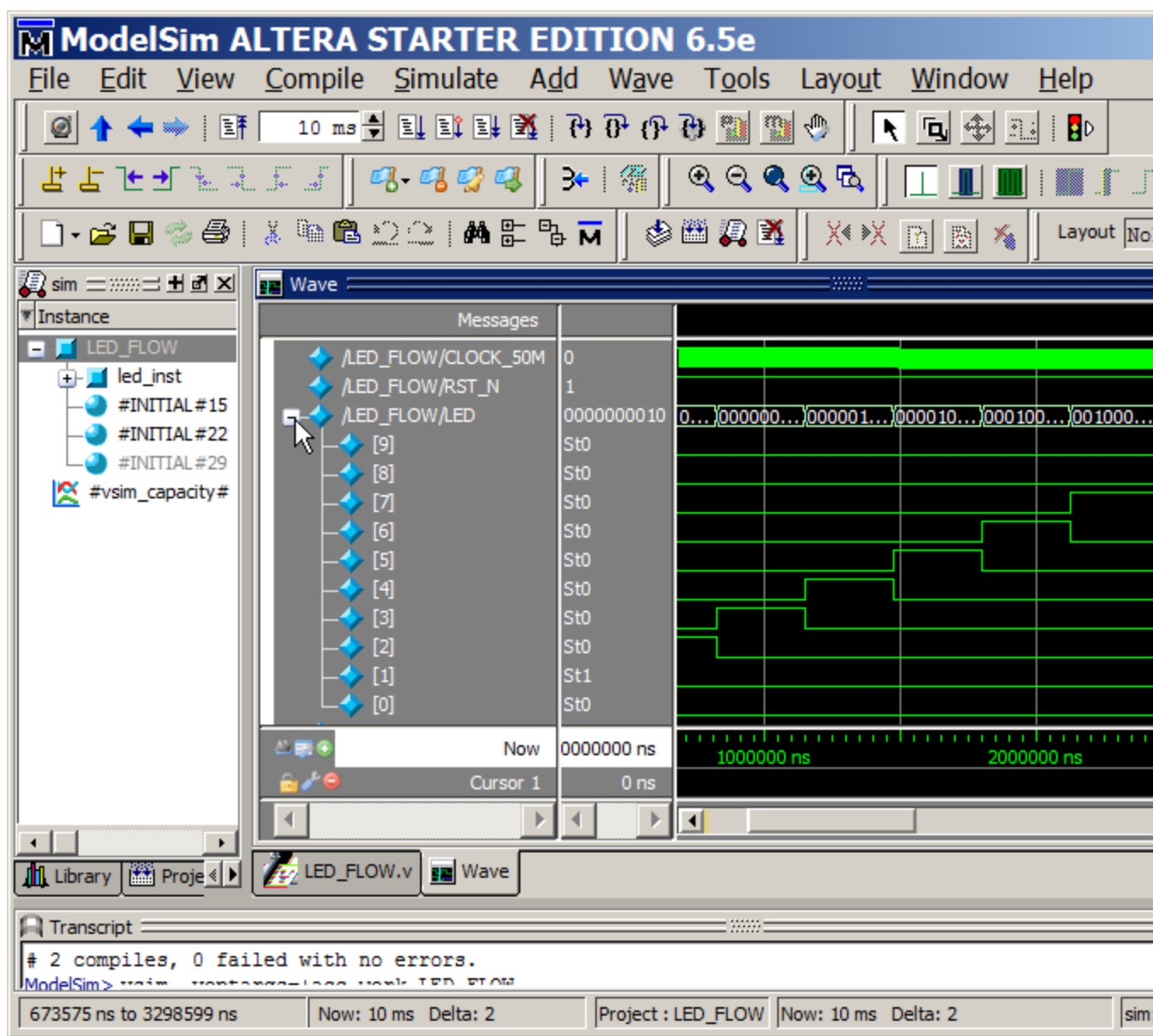


图 3.9 展开波形

通过放大/缩小波形，可以观察到 LED 的值在保持变化，即 LED 的时序效果。若将其移植到 Quartus II 中，适当配置后，经过综合、时序分析、引脚分配、配置及下载等，即可实现跑马灯的效果。