

# P8自动测评说明

## 硬件要求

### 添加测试用黑盒WRAPPER

- 把 **generate** 里的三个文件放到你ISE项目的根目录( `FIFO.ngc` , `P8_BLACKBOX.v` , `wrapper.ngc` ), 并把 `P8_BLACKBOX.v` 添加到工程中。
- 设置时钟主频为**10MHz** 一倍频是10M, 二倍是20M, 这个模块要接一倍频的。你可能需要重新生成时钟的ipcore。**注意重新生成时只需要修改clkout即可, clkin仍是25 (即在生成ipcore窗口的第一页不需要修改)**

The image shows the Xilinx Clocking Wizard interface. On the left, the IP Symbol is displayed, showing a block with multiple input and output ports. The main window is titled "Clocking Wizard" and shows the configuration for a component named "CLK\_A".

**Clocking Features:**

- ☒ Frequency synthesis
- ☒ Phase alignment (known phase relationship to input clock)
- ☐ Minimize power
- ☐ Dynamic phase shift
- ☐ Dynamic reconfiguration (in system output freq modification)

**Jitter Optimization:**

- ☒ Balanced
- ☐ Minimize output jitter (low clock jitter filtering)
- ☐ Maximize input jitter filtering (allow larger input jitter)

**Clock Manager Type:**

Mode:

- ☒ Auto Selection (Recommended: Wizard selects primitive)
- ☐ Manual Selection (User selects primitive)

**Input Jitter Unit:**

- ☒ UI
- ☐ PS

**Input Clock Information:**

| Input Clock | Input Freq (MHz) |                 | Input Jitter | Source                         |
|-------------|------------------|-----------------|--------------|--------------------------------|
|             | Value            | Valid Range     |              |                                |
| primary     | 25.000           | 5.000 - 500.000 | 0.010        | Single ended clock capable pin |

The IP Symbol shown on the left describes the input and output ports provided from the clocking network. Note that the names of these ports do not necessarily match the pins of any specific primitive.

Page 1 of 6

The phase is calculated relative to the active input clock.

| Output Clock                                 | Output Freq (MHz) |        | Phase (degrees) |        | Duty Cycle (%) |        | Drives | Use Fine Ps              |
|--|-------------------|--------|-----------------|--------|----------------|--------|--------|--------------------------|
|  | Requested         | Actual | Requested       | Actual | Requested      | Actual |        |                          |
| CLK_OUT1                                     | 10.000            | 10.000 | 0.000           | 0.000  | 50.000         | 50.0   | BUFG   | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> CLK_OUT2 | 20.000            | 20.000 | 0.000           | 0.000  | 50.000         | 50.0   | BUFG   | <input type="checkbox"/> |
| <input type="checkbox"/> CLK_OUT3            | 100.000           | N/A    | 0.000           | N/A    | 50.000         | N/A    | BUFG   | <input type="checkbox"/> |
| <input type="checkbox"/> CLK_OUT4            | 100.000           | N/A    | 0.000           | N/A    | 50.000         | N/A    | BUFG   | <input type="checkbox"/> |
| <input type="checkbox"/> CLK_OUT5            | 100.000           | N/A    | 0.000           | N/A    | 50.000         | N/A    | BUFG   | <input type="checkbox"/> |
| <input type="checkbox"/> CLK_OUT6            | 100.000           | N/A    | 0.000           | N/A    | 50.000         | N/A    | BUFG   | <input type="checkbox"/> |

- cpu顶层模块新增下面两个端口分别连接wrapper模块中的 rxd, TxD

```
input uart_rxd2,
output uart_txd2,
```

- 在你的项目中实例化P8\_BLACKBOX.v 中定义的wrapper模块，连接以下端口即可 (clk, reset, M\_addr, M\_WE, M\_WData, rxd, TxD)，其它端口不用连或写0即可，可参考下图，但请按自己的设计自行修改，不要完全照搬下图

```
clk      时钟信号10MHz
reset    复位信号
M_addr   DM的写地址
M_WE     DM的写使能
M_WData  DM的写数据
```

```

        wrapper WRAPPER(
            .clk(clk1),
            .reset(reset),
            .F_addr(0),
            .F_instr(0),
            .M_addr({DM_Address[31:2],2'b0}),
            .M_WE(DM_MemWrite),
            .M_RE(0),
            .M_WData(DM_WData),
            .M_RData(0),
            .user_F_addr(),
            .user_F_instr(),
            .user_M_addr(),
            .user_M_WE(),
            .user_M_RE(),
            .user_M_WData(),
            .user_M_RData(),
            .rxu(uart_rxd2),
            .TxD(uart_txd2),
            .txd_buffer()
        );

```

## 修改ucf文件

- 使用 `generate` 里的新的 pin.ucf

## 8个按钮中断要求

- 按钮中断要求如下：
  - 8个button均对应到同一个Hwint上，例如8个button均接到Hwint[4]上
  - 当一个button被按下时（脉冲为0），此时对应的Hwint应为1，并持续保持为1
  - 当上述button弹起来时（脉冲为1），此时对应的Hwint应为0
- 对于其他外设是否接入中断可自行设计

## save指令规范

- 对于DM，请务必保证为先写后读，并且读的数据为当前地址所在字的数据
- 对于sh和sb指令的行为，需要满足以下规范：
  - 对于sh和sb指令写的地址，需要以**字对齐**输出到WRAPPER的M\_addr上
  - 对于sh和sb指令写的值，需要将写完内存后把内存中当前地址所在字的值输出来，即WRAPPER的M\_WData端口需要接当前地址所在字的内存的值
  - 以上输出规范和P5P6的display输出规范一致，只是将display输出的形式转换为硬件实现

## 软件要求

- 测试的代码为 `P8_test_code.asm`
- IM导入的代码使用 `init.coe`
- 另外testbench代码 `mips_test.v` 文件可供参考，可以使用该文件进行仿真debug，确认没有bug后再上板子测试

## 测试原理介绍

- 本自动测评仅简单测试综合后的CPU是否具备基础功能

- 测试代码及测试原理大致如下：
  - 开始执行首先进入死循环，我们会按下一个button触发中断，你要保证你的button满足一些要求，见上述按钮中断要求。
  - 大约50ms后按钮弹起，在CPU退出中断后，我们会跳转到一个代码块起始地址，测试一个类似P6功能的代码块
  - 在代码块执行结束后，会再次重复上述操作进入中断切换到下一个代码块，依次类推

## 注意事项

---

- 目前已经开放了两个P8实验平台，一个是之前开放的自主测试平台，一个是刚刚开放的自动测评平台
- 不要把上述要求实现的cpu提交到正常实验平台上去
  - 即如果要在自动测评平台提交，则务必按上述要求实现好硬件
  - 如果要在自主测试平台进行自己测试，则不能将WRAPPER接入到CPU中，并且应该使用原来的ucf文件
  - 自动测评平台仅接受按上述要求操作实现的bit文件，自主测试平台仅接受之前要求的bit文件
  - 请务必确认好bit文件再提交，**否则可能烧坏板子**
- 加了这个模块的ucf文件和之前的ucf不同，请注意切换

**提示，在使用这个wrapper之后，行为仿真（综合前仿真）不好使了，可以使用综合后仿真来调试。**