

# 明德扬科技教育有限公司

## 异步时序之经典 CPU 接口传输练习说 明

官 网: [www.mdy-edu.com](http://www.mdy-edu.com)

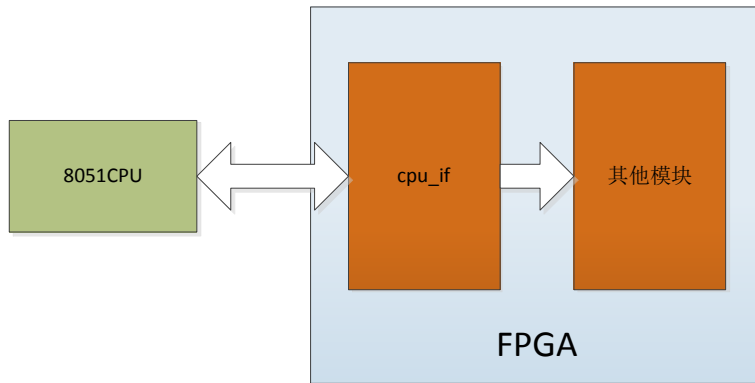
淘 宝: [mdy-edu.taobao.com](http://mdy-edu.taobao.com)

QQ 群: 97925396

QQ 咨询: 158063679

## 一、 练习内容及要求

在电路板上，通常有多个芯片，例如 8051+FPGA 组合。8051CPU 常用于配置 FPGA 的工作模式、读取 FPGA 的工作状态等。如下图所示。

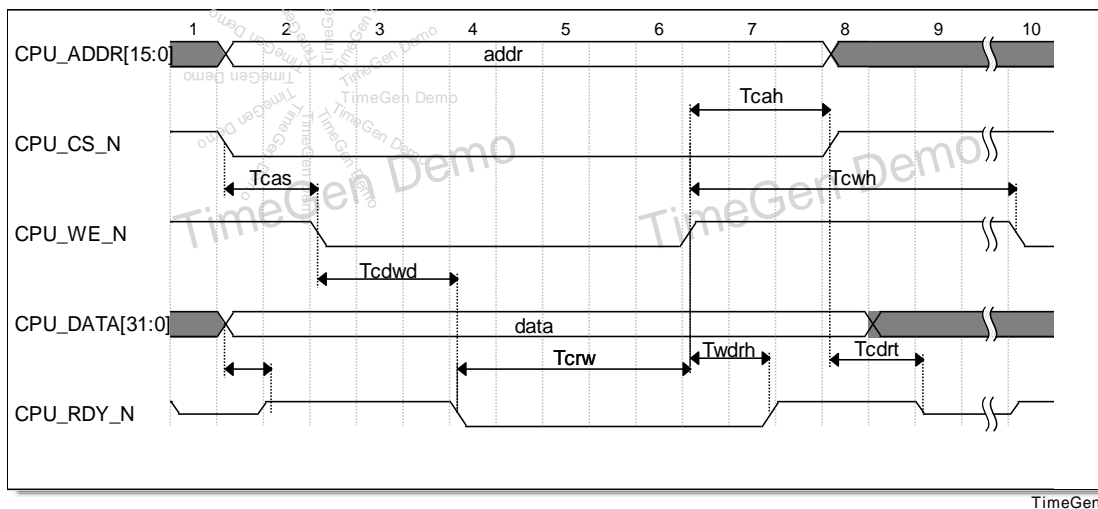


假设该 CPU 与 FPGA 通过异步接口信号通信，异步接口包括

- a 32 比特的数据总线 `cpu_data`
- b 16 比特的地址总线 `cpu_addr`
- c 读指示 `cpu_rd_n`
- d 写指示 `cpu_wr_n`
- e 片选指示 `cpu_cs_n`
- f 状态指示 `cpu_rdy_n`

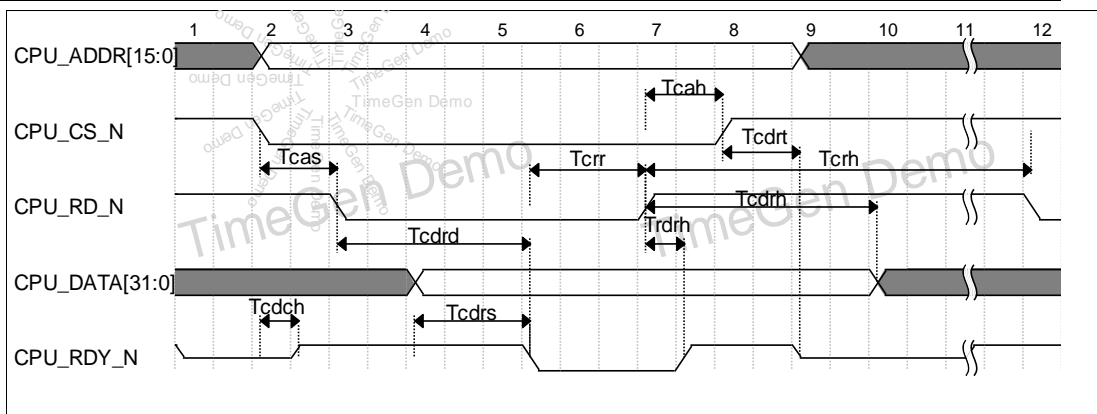


CPU 写接口时序（即 CPU 配置 FPGA，务必搞清楚每个步骤是谁主动的）：

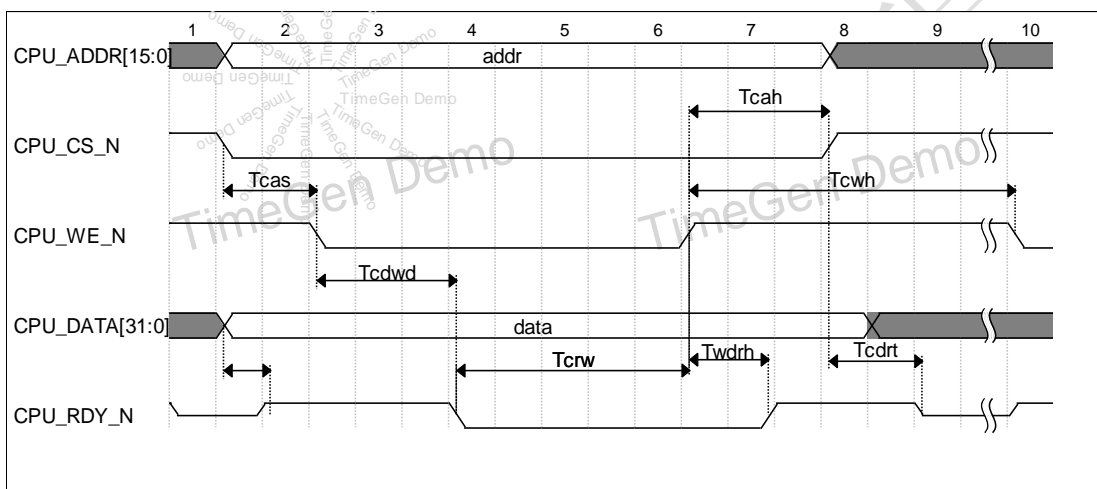


1. CPU 将 `cpu_cs_n` 信号拉低，并同时在 `cpu_addr` 中给出寄存器的地址，`cpu_data` 上给出要写的数据；
2. FPGA 识别出 `cpu_cs_n` 信号拉低后，将 `cpu_rdy_n` 信号由输出高阻变成输出高电平；
3. CPU 将 `cpu_we_n` 信号拉低，告知 FPGA 这是写操作；
4. FPGA 识别出写操作后，把 `cpu_data` 的值写到 `cpu_addr` 指定的寄存器上，然后 `cpu_rdy_n` 拉低，告知 CPU 将写完；
5. CPU 识别出 `cpu_rdy_n` 的下降沿，明白 FPGA 已经写完；
6. CPU 先后将 `cpu_we_n`；
7. FPGA 识别出 `cpu_we_n` 拉高后，将 `cpu_rdy_n` 信号拉高；
8. CPU 识别出 `cpu_rdy_n` 拉高后，将 `cpu_cs_n` 拉高；
9. FPGA 识别出 `cpu_cs_n` 拉高后，将 `cpu_rdy_n` 输出高阻。

注意：CPU 保证 `cpu_wr_n` 和 `cpu_rd_n` 不同时为低。



TimeGen



TimeGen

时序参数如下表所示（此表可下载，详见课程简介）。

Symbol	描述	Min (ns)	Max(ns)
Tcas	Address_chip select setup time	10	-
Tcah	Address_chip select hold time	10	-
Tcrh	Read high width	200	-
Tcdrh	Read data hold time after read de-assertion	30	120
Tcdch	Chip select assertion to ready high level	0	60
Tcdrd	Read data driving delay	80	200
Tcrr	Ready assertion to read de-assertion	10	-
Trdrh	Read de-assertion to ready de-assertion	10	80
Tcdrt	Chip select de-assertion to ready tri-state	0	100
Tcwh	Write high width	200	-
Tcdwd	Write assertion to ready assertion	80	200
Tcrw	Ready assertion to write de-assertion	10	-
Tcdrs		10	
Twdrh	Write de-assertion to ready de-assertion	0	80

本练习设计 `cpu_if` 模块。该模块具有如下功能

1. CPU 通过读时序，可以读到 FPGA 内部的寄存器
  - a 读地址为 0 时，将 `cfg_chan` 的值返回给 CPU。
  - b 读地址为 1 时，将 `cfg_mode` 的值返回给 CPU 低 8 位，其他位补 0。
  - c 读地址为 2 时，将 `sta_fpga` 的值返回给 CPU 低 16 位，高位补 0。
  - d 读地址为其他时，返回给 CPU 的值为 0。
2. CPU 通过写时序，可以配置 FPGA 内部的寄存器
  - a 写地址为 0 时，将 CPU 总线上的值赋给 `cfg_chan`。
  - b 写地址为 1 时，将 CPU 总线上的低 8 位赋给 `cfg_mode`。
  - c 写地址为其他时，无任何变化。

以上功能，可以用寄存器列表来表示。

## 二、寄存器列表

RW: 可读可写; RO: 只读; WO: 只写; RC: 读清

Bits	Field	ADDR	Access Type	Description	Reset Val
31: 0	<code>cfg_chan</code>	16'h0000	RW	FPGA 的工作通道	0

Bits	Field	ADDR	Access Type	Description	Reset Val
31:8	RESERVED	16'h0001	RO	保留	0
7: 0	<code>cfg_mode</code>	16'h0001	RW	FPGA 的工作模式	0

Bits	Field	ADDR	Access Type	Description	Reset Val
31:16	RESERVED	16'h0002	RO	保留	0
15: 0	<code>sta_fpga</code>	16'h0002	RO	FPGA 的工作状态	0

## 三、信号列表

信号名	I/O	位宽	说明
<code>clk</code>	I	1	工作时钟，50MHz
<code>rst_n</code>	I	1	复位信号
<code>cpu_data_w</code>	O	32	写数据总线，FPGA 往 CPU 送的数据。连接到三态门上
<code>cpu_data_w_e</code>	O	1	写数据使能，连接到三态门上。该值为 1 时，表示打开三态

			门，将 <code>cpu_data_w</code> 的值送到总线上。
<code>cpu_data_r</code>	I	32	从数据总线三态门获取的数据
<code>cpu_addr</code>	I	16	地址总线
<code>cpu_rd_n</code>	I	1	读指示
<code>cpu_wr_n</code>	I	1	写指示
<code>cpu_cs_n</code>	I	1	片选指示
<code>cpu_rdy_w</code>	O	1	状态指示数据信号，连接到三态门上
<code>cpu_rdy_w_e</code>	O	1	写状态指示使能信号，连接到三态门上。该值为 1 时，表示打开三态门，将 <code>cpu_rdy_w</code> 的值送到总线上。
<code>cfg_mode</code>	O	8	发给其他模块使用的工作模式指示信号。
<code>cfg_chan</code>	O	32	发给其他模块使用的工作通道指示信号
<code>sta_fpga</code>	I	16	从其他模块过来的工作状态指示信号

### 三、学习建议

- 1、按照功能要求，思考、设计出代码。
- 2、编写测试文件，对代码进行仿真。建议包含如下测试点：
  - a `cpu_rd_n` 在时钟上升沿前一点变低，确认时序满足要求；
  - b `cpu_rd_n` 在时钟上升沿后一点变低，确认时序满足要求；
  - c `cpu_we_n` 在时钟上升沿前一点变低，确认时序满足要求；
  - d `cpu_we_n` 在时钟上升沿后一点变低，确认时序满足要求；
  - e 连续读写、读读、写写之间的间隔为 200ns，读写正确；
  - f `cpu_cs_n` 为 1，其他信号时序都正确，看是能正确读写，特别要注意三态门不能使能。
- 3、学习配套的设计思路视频，找出自己思路的异同，选择较好的思路
- 4、按照最新的设计思路，再实现一次。