



明德扬
科技·教育

异步时序之经典CPU接口传输 练习时序参数



点透学习误区 拨出设计精髓

主 讲：潘文明

明德扬科教

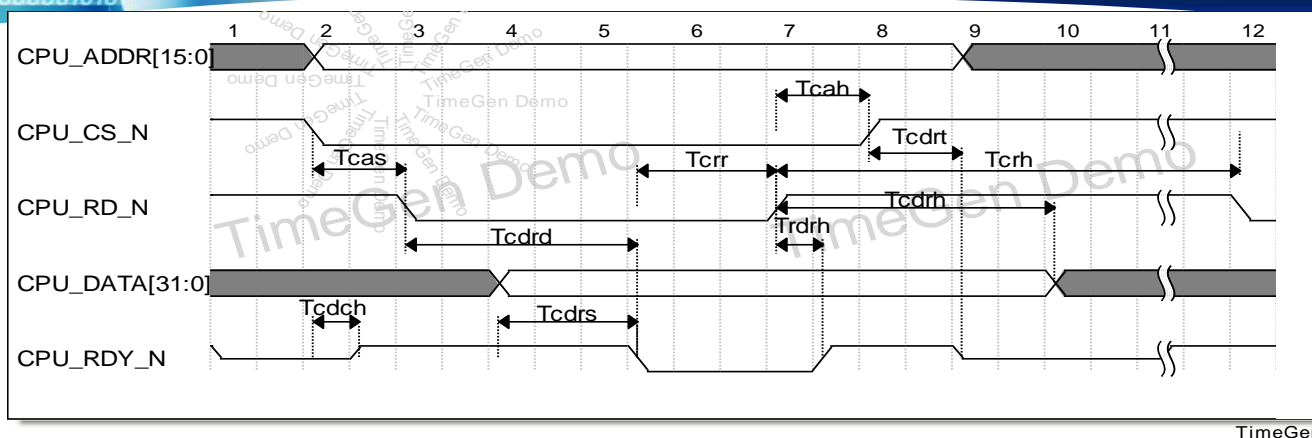


QQ群: 97925396

官 网: <http://www.mdy-edu.com>

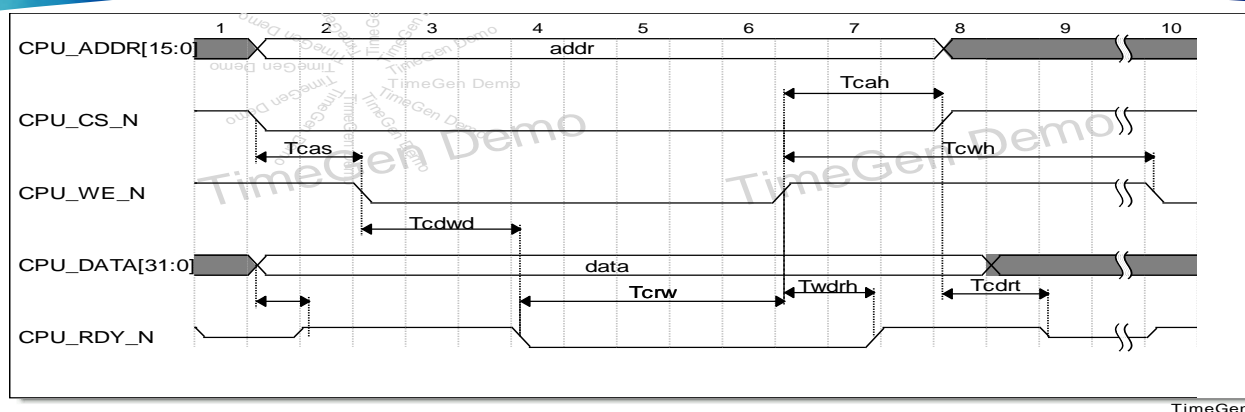
淘 宝: <http://mdy-edu.taobao.com>

一、读时序



1. CPU将cpu_cs_n信号拉低，同时在cpu_addr中给出寄存器的地址；
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_rd_n拉低，告知FPGA这是读操作；
4. FPGA识别出读操作后，把寄存器数据输出到cpu_data上，然后将cpu_rdy_n输出低电平，告知CPU数据已经准备好；
5. CPU识别出cpu_rdy_n的下降沿，将会读取cpu_data上的值；
6. CPU将cpu_rd_n拉高；
7. FPGA识别出cpu_rd_n拉高后，知道cpu已经读走了数据，则将cpu_rdy_n信号拉高；
8. CPU识别出cpu_rdy_n信号拉高后，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。

二、写时序

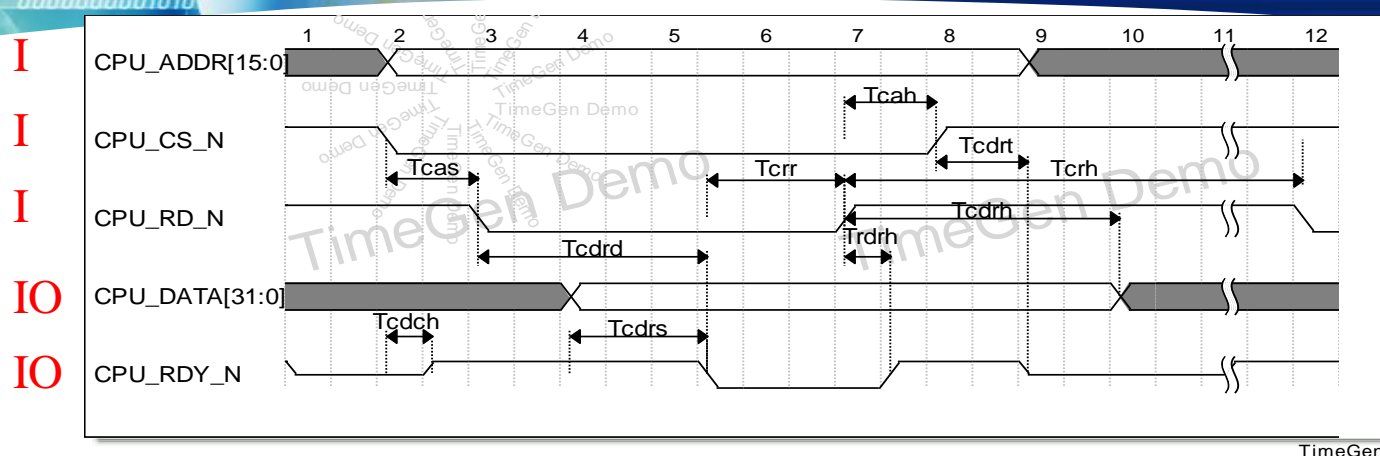


1. CPU将cpu_cs_n信号拉低，并同时在cpu_addr中给出寄存器的地址，cpu_data上给出要写的数据；
2. FPGA识别出cpu_cs_n信号拉低后，将cpu_rdy_n信号由输出高阻变成输出高电平；
3. CPU将cpu_we_n信号拉低，告知FPGA这是写操作；
4. FPGA识别出写操作后，把cpu_data的值写到cpu_addr指定的寄存器上，然后cpu_rdy_n拉低，告知CPU将写完；
5. CPU识别出cpu_rdy_n的下降沿，明白FPGA已经写完；
6. CPU将cpu_we_n拉高；
7. FPGA识别出cpu_we_n拉高后，将cpu_rdy_n信号拉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

二、观察过程

1. 对于设计模块，哪些是输入；哪些是输出；哪些是双向
2. 分清每个步骤的控制方
3. 分清每个步骤交流的信息
4. 看参数表
5. 分清楚哪些参数是对方保证，哪些参数是设计模块要保证
6. 设计的原则：对于满足参数的任何输入（无论怎么变），都保证能接收正确；对于输出，在满足时序要求前提下简化设计。

二、读时序观察

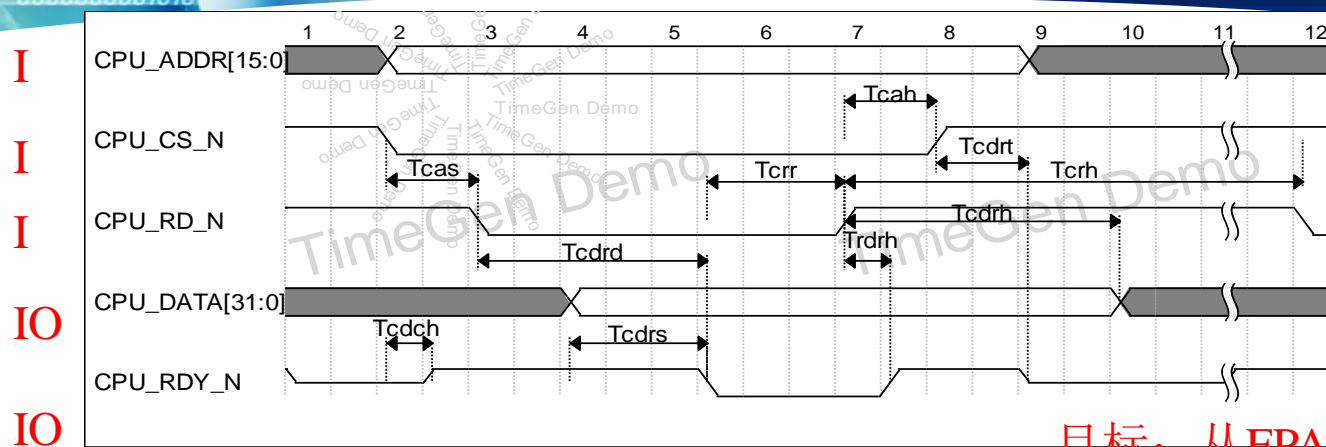


1. CPU将cpu_cs_n信号拉低，同时在cpu_addr中给出寄存器的地址；
2. FPGA识别出cpu_cs_n信号拉低后，将cpu_rdy_n信号由输出高阻变成输出高电平；
3. CPU将cpu_rd_n拉低，告知FPGA这是读操作；
4. FPGA识别出读操作后，把寄存器数据输出到cpu_data上，然后将cpu_rdy_n输出低电平，告知CPU数据已经准备好；
5. CPU识别出cpu_rdy_n的下降沿，将会读取cpu_data上的值；
6. CPU先后将cpu_rd_n和cpu_cs_n信号拉高；
7. FPGA识别出cpu_rd_n拉高后，知道cpu已经读走了数据，则将cpu_rdy_n信号拉高；
8. CPU识别出cpu_rdy_n信号拉高后，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。

二、观察过程

1. 对于设计模块，哪些是输入；哪些是输出；哪些是双向
2. 分清每个步骤的控制方
3. 分清每个步骤交流的信息
4. 看参数表
5. 分清楚哪些参数是对方保证，哪些参数是设计模块要保证
6. 设计的原则：对于满足参数的任何输入（无论怎么变），都保证能接收正确；对于输出，在满足时序要求前提下简化设计。

二、读时序观察



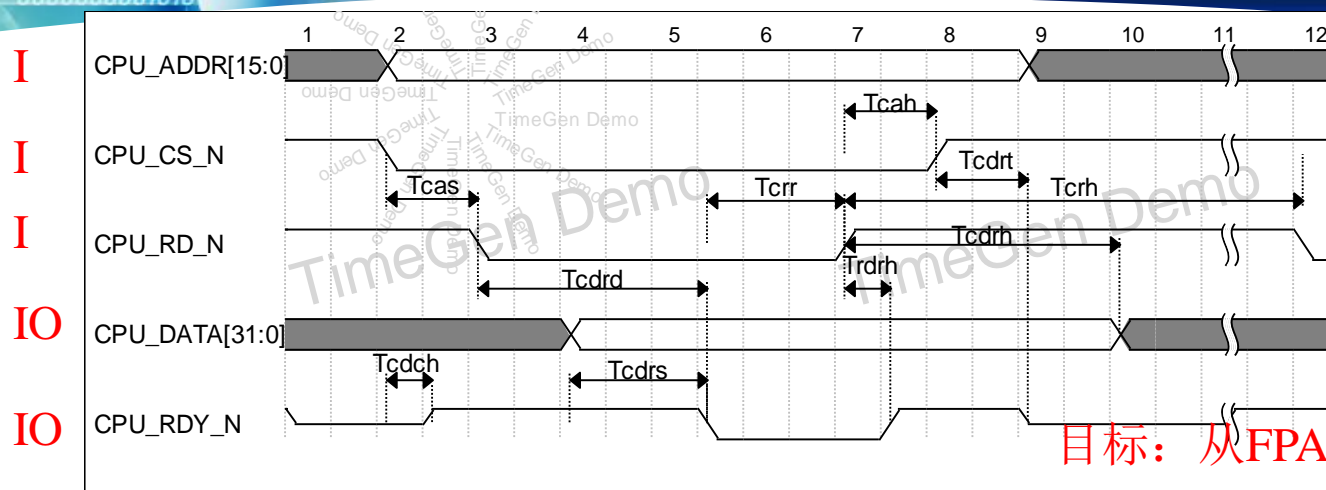
目标：从FPGA中得到一个数！

1. CPU将cpu_cs_n信号拉低，同时在cpu_addr中给出寄存器的地址； CPU
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； FPGA
3. CPU将cpu_rd_n拉低，告知FPGA这是读操作； CPU
4. FPGA识别出读操作后，把寄存器数据输出到cpu_data上，然后将cpu_rdy_n输出低电平，告知CPU数据已经准备好； FPGA
5. CPU识别出cpu_rdy_n的下降沿，将会读取cpu_data上的值； CPU
6. CPU先后将cpu_rd_n和cpu_cs_n信号拉高； CPU
7. FPGA识别出cpu_rd_n拉高后，知道cpu已经读走了数据，则将cpu_rdy_n信号拉高； FPGA
8. CPU识别出cpu_rdy_n信号拉高后，将cpu_cs_n拉高； CPU
9. FPGA识别出cpu_cs_n拉高后，将cpu_rdy_n输出高阻。 FPGA

二、观察过程

1. 对于设计模块，哪些是输入；哪些是输出；哪些是双向
2. 分清每个步骤的控制方
3. 分清每个步骤交流的信息
4. 看参数表
5. 分清楚哪些参数是对方保证，哪些参数是设计模块要保证
6. 设计的原则：对于满足参数的任何输入（无论怎么变），都保证能接收正确；对于输出，在满足时序要求前提下简化设计。

二、读时序观察



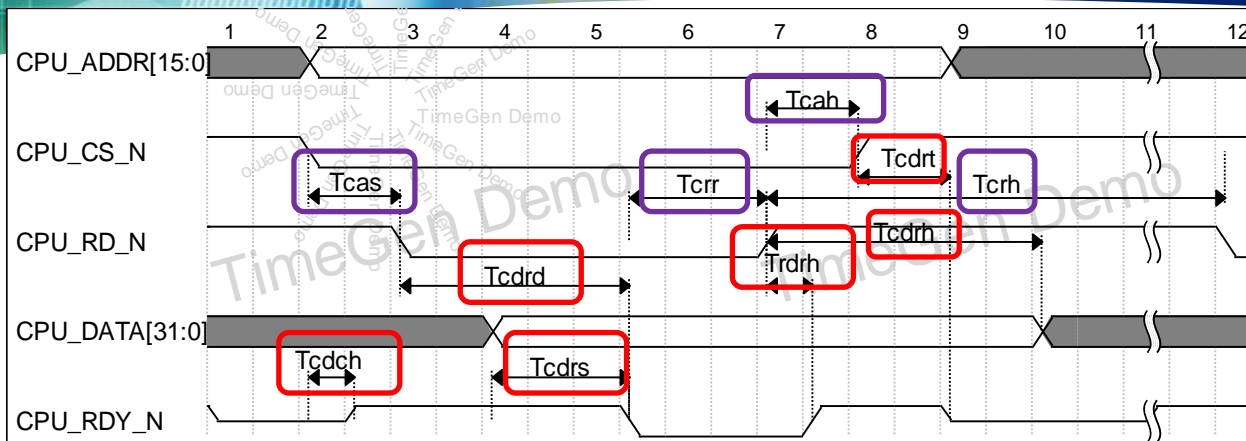
目标：从FPGA中得到一个数！

1. CPU将cpu_cs_n信号拉低，同时在cpu_addr中给出寄存器的地址；
FPGA知道选中和地址
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；
CPU选中了FPGA
3. CPU将cpu_rd_n拉低，告知FPGA这是读操作；
FPGA知道这是读操作
4. FPGA识别出读操作后，把寄存器数据输出到cpu_data上，然后将cpu_rdy_n输出低电平，告知CPU数据已经准备好；
CPU知道读完成
5. CPU识别出cpu_rdy_n的下降沿，将会读取cpu_data上的值；
6. CPU先后将cpu_rd_n和cpu_cs_n信号拉高；
FPGA知道CPU要结束操作
7. FPGA识别出cpu_rd_n拉高后，知道cpu已经读走了数据，则将cpu_rdy_n信号拉高；
CPU知道读结束
8. CPU识别出cpu_rdy_n信号拉高后，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。

二、观察过程

1. 对于设计模块，哪些是输入；哪些是输出；哪些是双向
2. 分清每个步骤的控制方
3. 分清每个步骤交流的信息
4. 看参数表
5. 分清楚哪些参数是对方保证，哪些参数是设计模块要保证
6. 设计的原则：对于满足参数的任何输入（无论怎么变），都保证能接收正确；对于输出，在满足时序要求前提下简化设计。

二、读时序观察

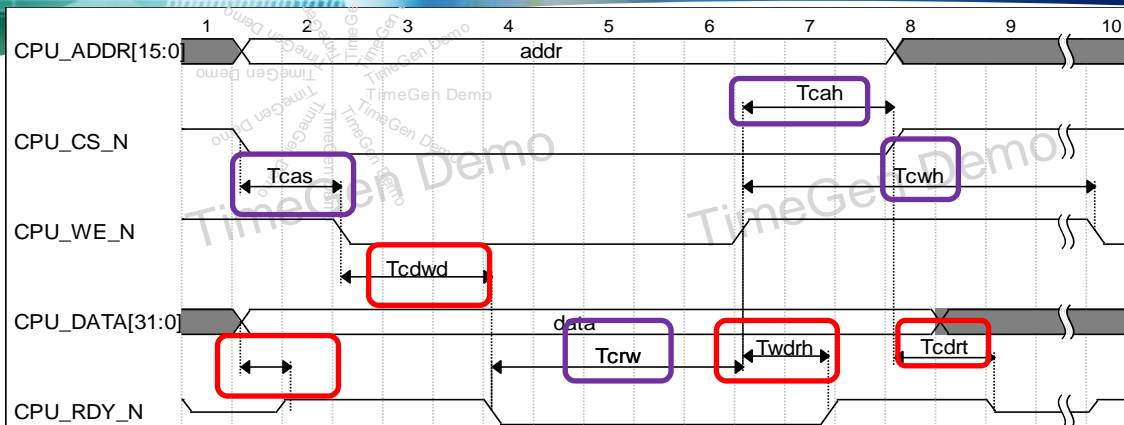


Symbol	描述	Min (ns)	Max(ns)
Tcas	Address_chip select setup time	10	-
Tcah	Address_chip select hold time	10	-
Tcrh	Read high width	200	-
Tcdrh	Read data hold time after read de-assertion	30	120
Tcdch	Chip select assertion to ready high level	0	60
Tcdrd	Read data driving delay	80	200
Tcrr	Ready assertion 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

1. Tcas: cs=0到rd=0至少有10ns
2. Tcrr: rdy=0到rd=1至少有10ns
3. Tcah: rd=1到cs=1至少有10ns
4. Tcrh: 两次读之间至少有200ns

1. Tcdch: cs=0到rdy=1要在0~60ns范围
2. Tcdrd: rd=0到rdy=0在80~200ns范围
3. Tcdrs: 数据在总线至rdy=0必须大于10ns
4. Trdrh: rd=1到rdy=1在10~80ns范围
5. Tcdrt: cs=1到rdy=z在0~100ns范围
6. Tcdrh: rd=1到数据无效在30~120ns范围

二、写时序观察



1. Tcas: cs=0到we=0至少有10ns
2. Tcrw: rdy=0到we=1至少有10ns
3. Tcah: we=1到cs=1至少有10ns
4. Tcwh: 两次写之间至少有200ns

Symbol	描述	Min (ns)	Max(ns)
Tcas	Address_chip select setuptime	10	-
Tcah	Address_chip select holdtime	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

1. Tcdch: cs=0到rdy=1要在0~60ns范围
2. Tcdwd: we=0到rdy=0在80~200ns
3. Tcdrs: 数据在总线至rdy=0必须大于10ns
4. Twdrh: we=1到rdy=1在10~80ns范围
5. Tcdrt: cs=1到rdy=z在0~100ns范围

二、观察过程

1. 对于设计模块，哪些是输入；哪些是输出；哪些是双向
2. 分清每个步骤的控制方
3. 分清每个步骤交流的信息
4. 看参数表
5. 分清楚哪些参数是对方保证，哪些参数是设计模块要保证
6. 设计的原则：对于满足参数的任何输入（无论怎么变），都保证能接收正确；对于输出，在满足时序要求前提下简化设计。



QQ群: 97925396

官 网: <http://www.mdy-edu.com>

淘 宝: <http://mdy-edu.taobao.com>



Thank You !

