



清华大学  
Tsinghua University

# THCO MIPS指令计算机 系统设计和实现

## Project 4 & 5

刘卫东

计算机科学与技术系

# 内容提要



清华大学  
Tsinghua University

- ✚我们要做什么？
- ✚我们已经有了什么？
- ✚我们可以怎样去做？
- ✚我们还能做什么？
- ✚我们最后要交什么？
- ✚具体时间安排

# 我们要做什么？



清华大学  
Tsinghua University

## 奋战三星期

## 做台计算机

# 我们要做什么？



## ❖ 实验任务

- ❑ 多周期CPU或支持指令流水的CPU（二选一）
- ❑ 使用基本存储和扩展存储以及输入/输出
- ❑ 进行扩展

## ❖ 实验目标

- ❑ 能运行监控程序,并在监控程序中运行PROJECT 1的程序
- ❑ 有可供演示的应用程序

# 我们已经有了什么？



## THINPAD 硬件平台

- ❑ FPGA CPU
- ❑ RAM Memory
- ❑ FLASH Memory
- ❑ UART/USB/PS2/VGA/

## THCO MIPS 指令系统

## 监控程序等辅助软件

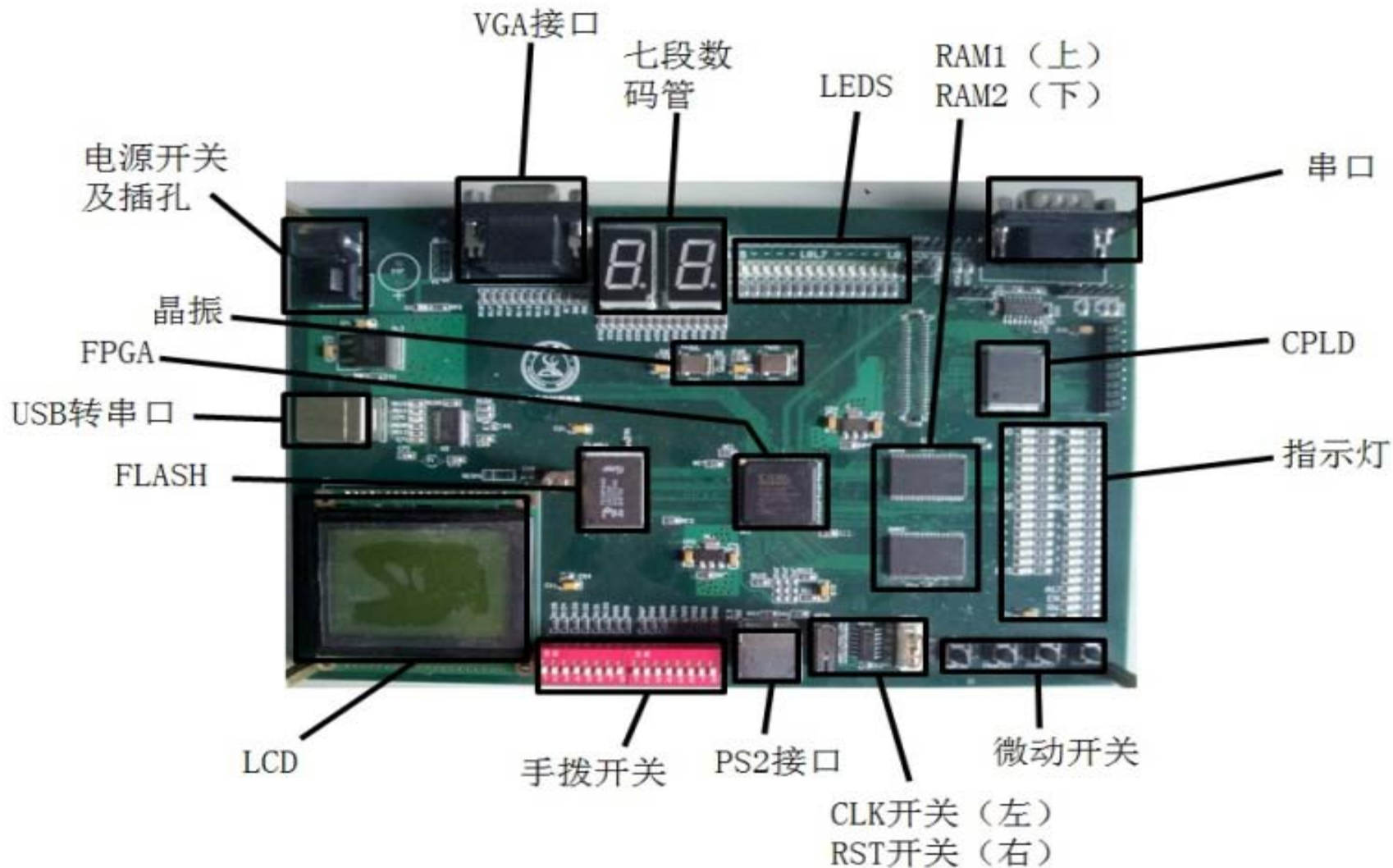
## Project1 & Project2 & Project3

## 教材和实验指导书

# 教学计算机硬件平台



清华大学  
Tsinghua University



# 教学机硬件基本组成



## ✚ CPU

- ✚ FPGA

## ✚ Memory

- ✚ 基本: 256KW Ram

- ✚ 扩展: 256KW Ram

## ✚ Bus

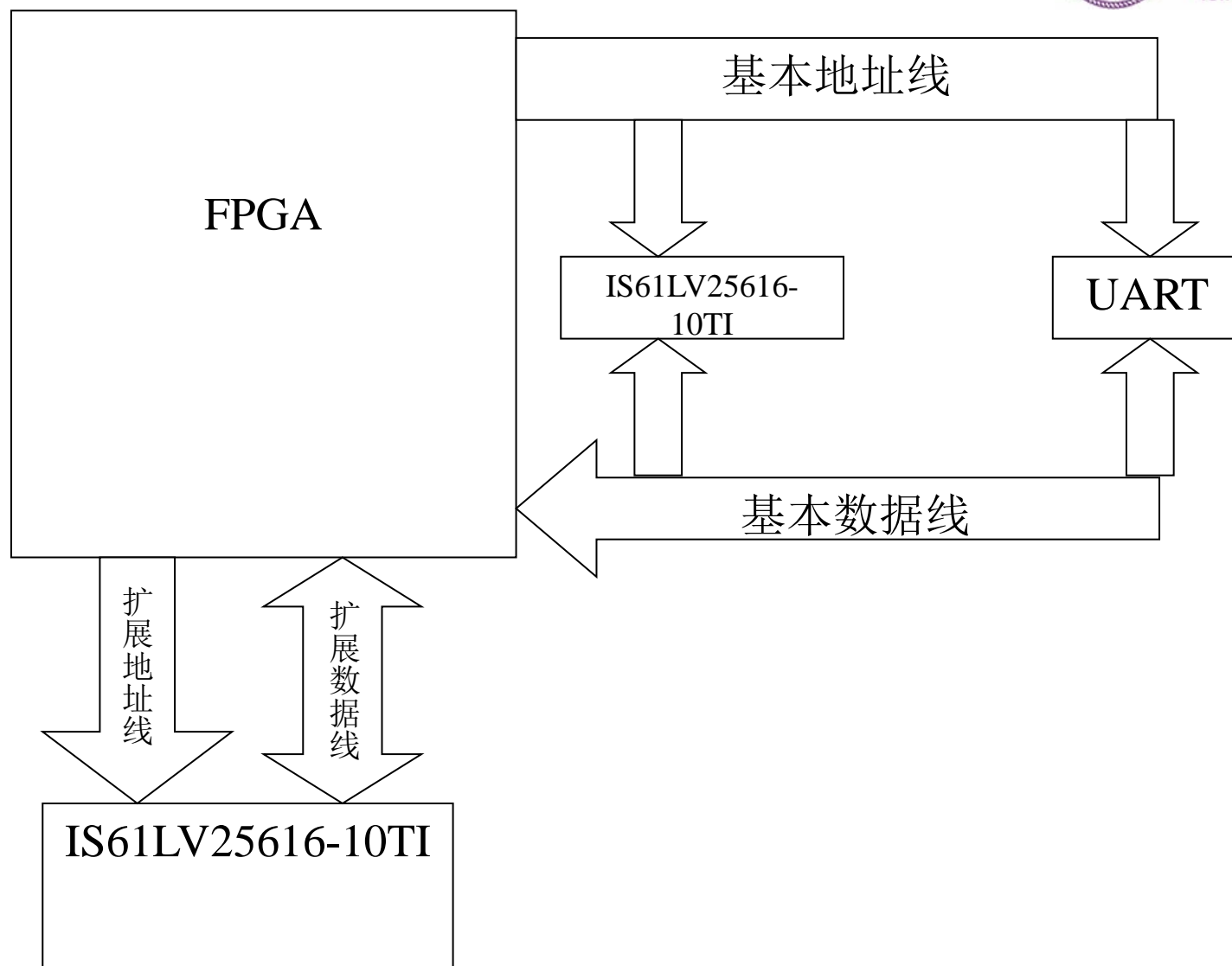
- ✚ 基本: 数据16bits/地址18bits/控制 3bits

- ✚ 扩展: 数据16bits /地址18bits /控制3bits

## ✚ Device

- ✚ UART

# 结构





✿ Xilinx公司的SPARTAN-3E系列芯片

✿ 型号：XC3S1200EFGG320

▣ 120万门容量

▣ 2168个CLB

▣ 504Kb的RAM块

▣ 320脚的FBGA封装形式

# FPGA管脚



- ⊕ 连接基本存储
  - ⊠ 地址18位
  - ⊠ 数据16位
- ⊕ 连接扩展存储
  - ⊠ 地址18位
  - ⊠ 数据16位
- ⊕ 总线控制信号
  - ⊠ 各3位
- ⊕ 指示灯
- ⊕ FLASH
- ⊕ PS2
- ⊕ VGA
- ⊕ 具体见实验指导书

# CPU功能



## 数据通路

.....

## 控制器

.....

## 框架结构

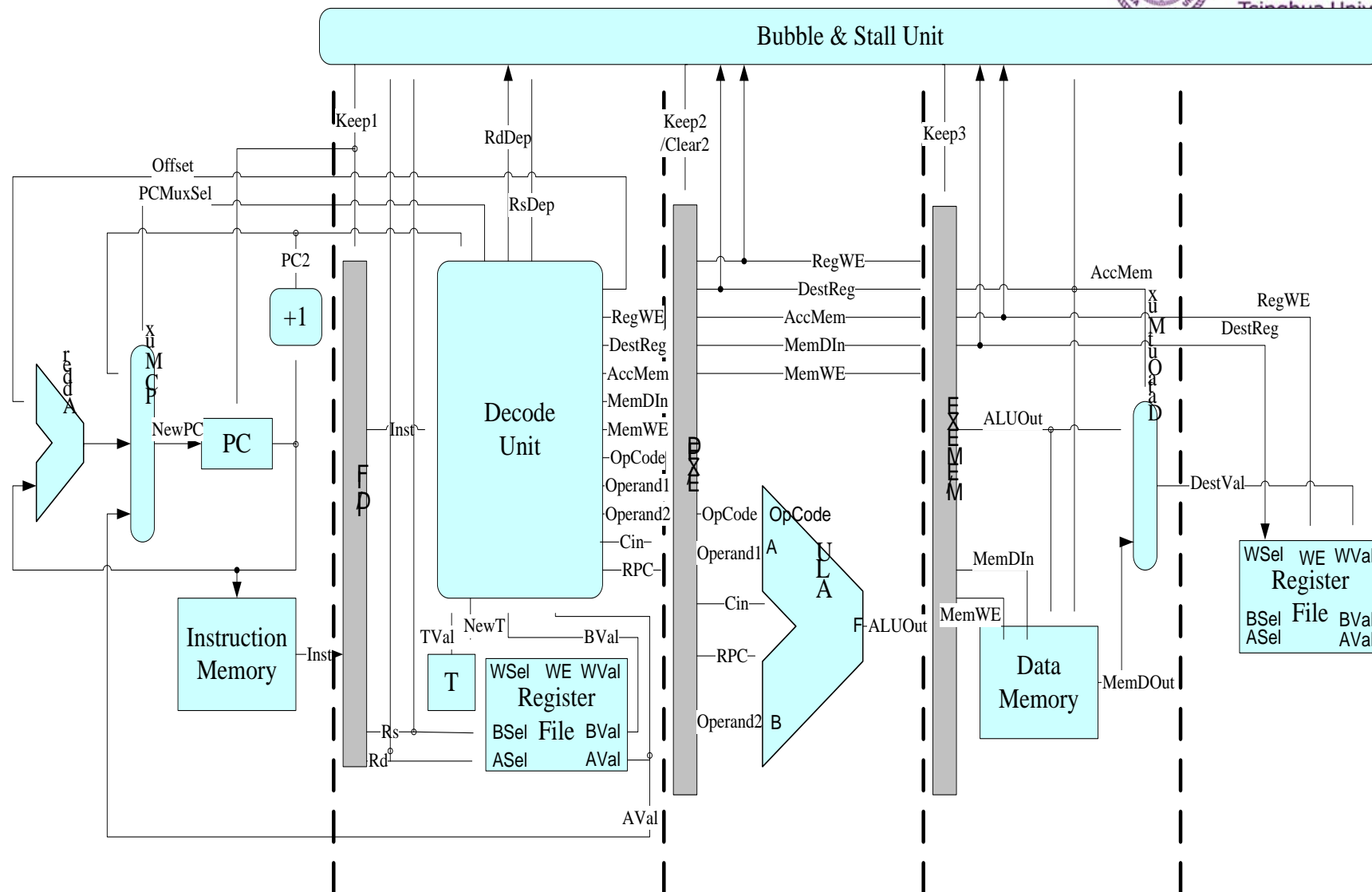
ALU

Register Files

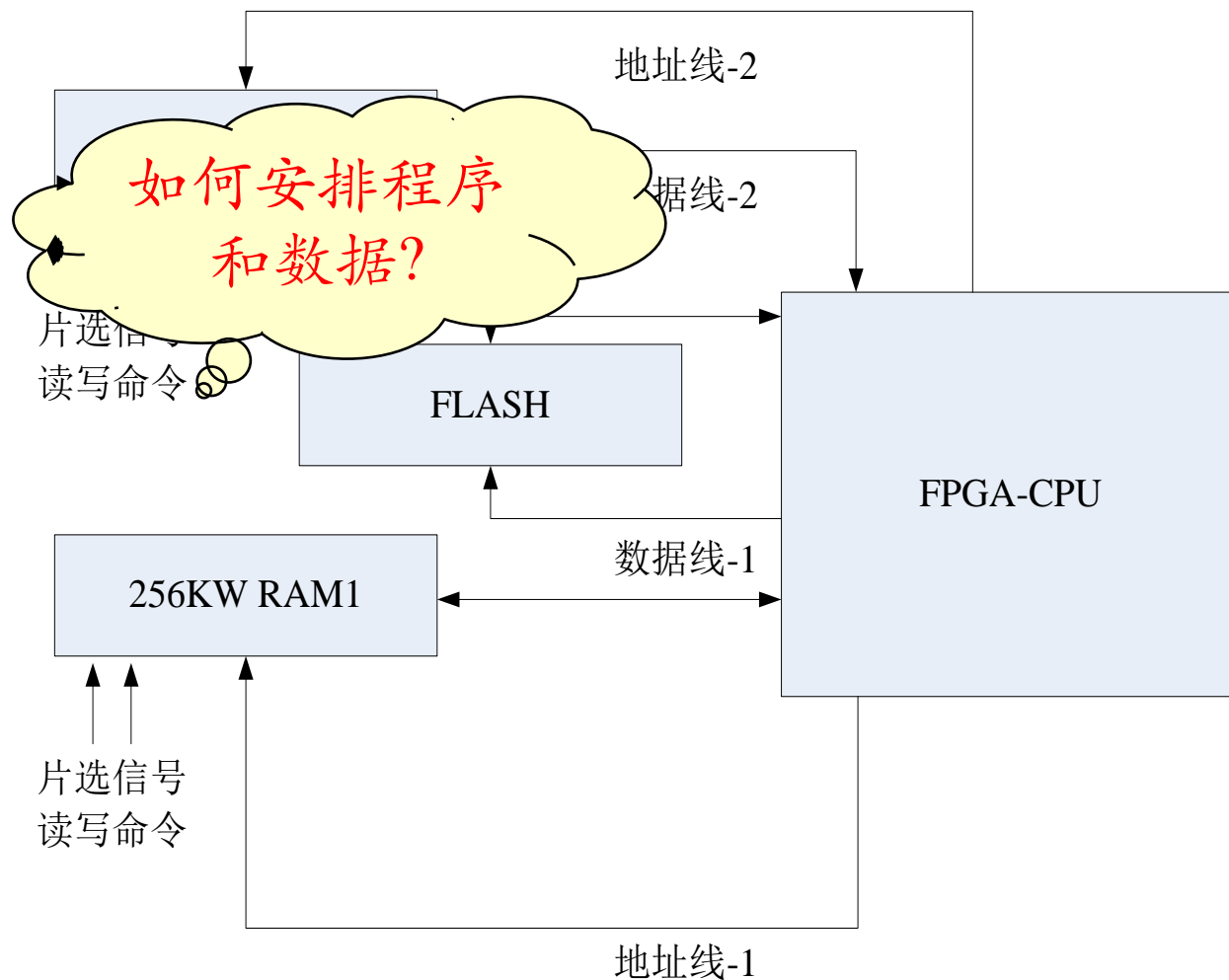
MEM Interface

Decoder

# 框架结构



# THINPAD存储器



# 监控程序划分的地址段



功能区	地址段	说明
系统程序区	0x0000~0x3FFF (32K)	存放监控程序
用户程序区	0x4000~0x7FFF (32K)	存放用户程序
系统数据区	0x8000~0xBEFF	监控程序使用的数据区
Com1数据端口/命令端口	0xBF00~0xBF01	第1个串口的端口
Com2数据端口/命令端口	0xBF02~0xBF03	第2个串口的端口
预留给其他接口	0xBF04~0xBF0F	保留
系统堆栈区	0xBF10~0xBFFF	用于系统堆栈
用户数据区	0xC000~0xFFFF	用户程序使用的数据区

# 程序如何装入？



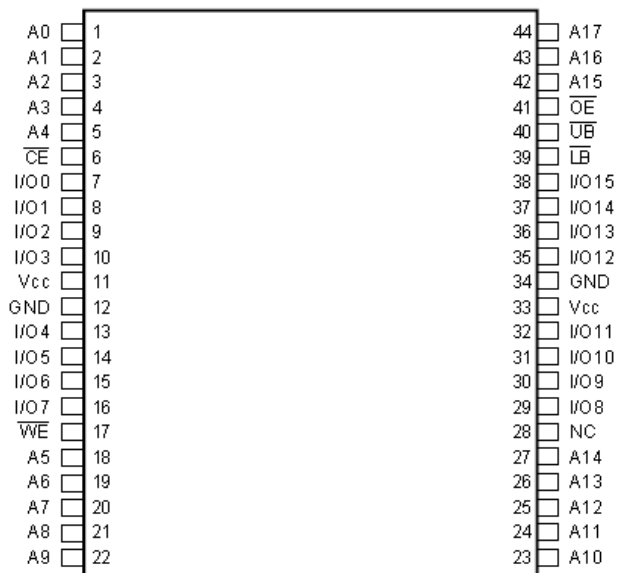
## ❁ 程序装入到FLASH中

- ❏ 采用提供的软件装入
- ❏ CPU首先将其boot到RAM1中
- ❏ 再从RAM1中运行

## ❁ 程序直接装入到RAM1中

- ❏ 直接用软件装入到RAM1中

# 内存访问



## 控制信号

$\square$  /CE、/WE、/OE

## 访问时序

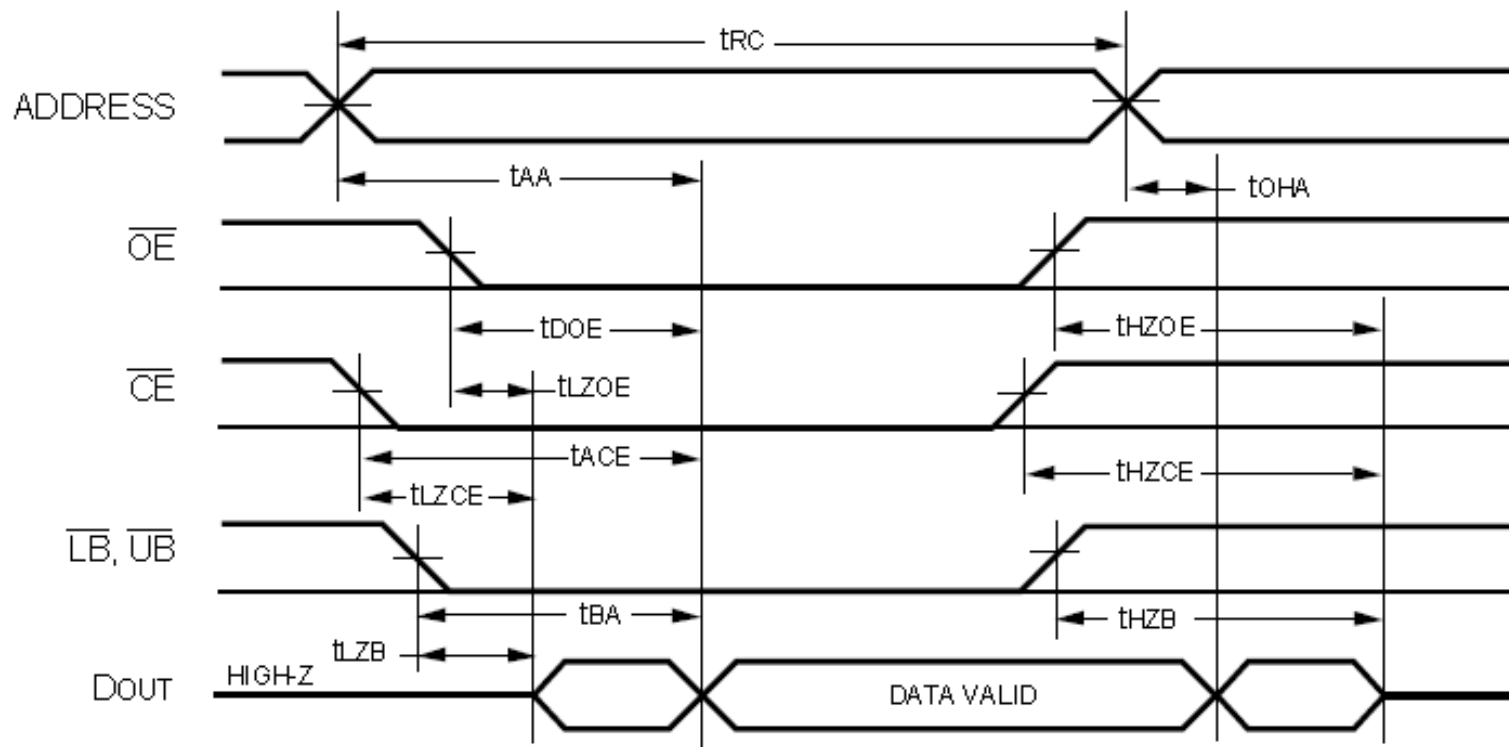
$\square$  如何保证?

TRUTH TABLE

Mode	$\overline{WE}$	$\overline{CE}$	$\overline{OE}$	$\overline{LB}$	$\overline{UB}$	I/O PIN		Vcc Current
						I/O0-I/O7	I/O8-I/O15	
Not Selected	X	H	X	X	X	High-Z	High-Z	Isb1, Isb2
Output Disabled	H	L	H	X	X	High-Z	High-Z	Icc
	X	L	X	H	H	High-Z	High-Z	
Read	H	L	L	L	H	DOUT	High-Z	Icc
	H	L	L	H	L	High-Z	DOUT	
	H	L	L	L	L	DOUT	DOUT	
Write	L	L	X	L	H	DIN	High-Z	Icc
	L	L	X	H	L	High-Z	DIN	
	L	L	X	L	L	DIN	DIN	



# 内存读时序



# 内存读参数



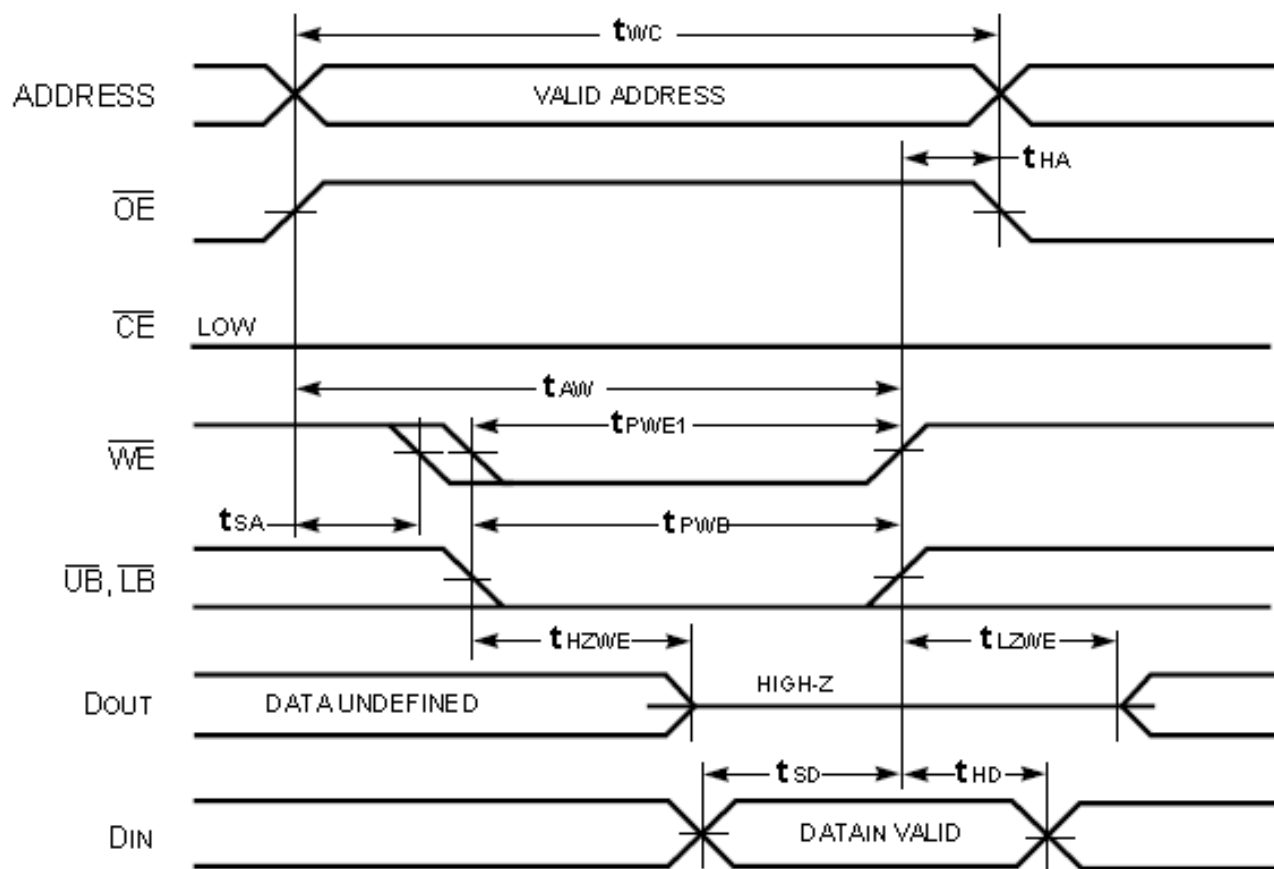
READ CYCLE SWITCHING CHARACTERISTICS<sup>(1)</sup> (Over Operating Range)

Symbol	Parameter	-8		-10		-12		-15		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
t <sub>RC</sub>	Read Cycle Time	8	—	10	—	12	—	15	—	ns
t <sub>AA</sub>	Address Access Time	—	8	—	10	—	12	—	15	ns
t <sub>OHA</sub>	Output Hold Time	3	—	3	—	3	—	3	—	ns
t <sub>ACE</sub>	$\overline{\text{CE}}$ Access Time	—	8	—	10	—	12	—	15	ns
t <sub>DOE</sub>	$\overline{\text{OE}}$ Access Time	—	4	—	5	—	6	—	7	ns
t <sub>HZOE</sub> <sup>(2)</sup>	$\overline{\text{OE}}$ to High-Z Output	0	4	—	5	—	6	0	6	ns
t <sub>LZOE</sub> <sup>(2)</sup>	$\overline{\text{OE}}$ to Low-Z Output	0	—	0	—	0	—	0	—	ns
t <sub>HZCE</sub> <sup>(2)</sup>	$\overline{\text{CE}}$ to High-Z Output	0	4	0	5	0	6	0	6	ns
t <sub>LZCE</sub> <sup>(2)</sup>	$\overline{\text{CE}}$ to Low-Z Output	3	—	3	—	3	—	3	—	ns
t <sub>BA</sub>	$\overline{\text{LB}}, \overline{\text{UB}}$ Access Time	—	4	—	5	—	6	—	7	ns
t <sub>HZB</sub>	$\overline{\text{LB}}, \overline{\text{UB}}$ to High-Z Output	0	4	0	5	0	6	0	6	ns
t <sub>LZB</sub>	$\overline{\text{LB}}, \overline{\text{UB}}$ to Low-Z Output	0	—	0	—	0	—	0	—	ns

# 内存写时序



WRITE CYCLE NO. 2 ( $\overline{WE}$  Controlled.  $\overline{OE}$  is HIGH During Write Cycle) <sup>(1,2)</sup>



# 内存写参数



WRITE CYCLE SWITCHING CHARACTERISTICS<sup>(1,3)</sup> (Over Operating Range)

Symbol	Parameter	-8		-10		-12		-15		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
t <sub>WC</sub>	Write Cycle Time	8	—	10	—	12	—	15	—	ns
t <sub>SC</sub>	$\overline{CE}$ to Write End	7	—	8	—	9	—	10	—	ns
t <sub>AW</sub>	Address Setup Time to Write End	7	—	8	—	9	—	10	—	ns
t <sub>HA</sub>	Address Hold from Write End	0	—	0	—	0	—	0	—	ns
t <sub>SA</sub>	Address Setup Time	0	—	0	—	0	—	0	—	ns
t <sub>PWB</sub>	$\overline{LB}$ , $\overline{UB}$ Valid to End of Write	7	—	8	—	9	—	10	—	ns
t <sub>PWE</sub>	$\overline{WE}$ Pulse Width	7	—	8	—	9	—	10	—	ns
t <sub>SD</sub>	Data Setup to Write End	4.5	—	5	—	6	—	7	—	ns
t <sub>HD</sub>	Data Hold from Write End	0	—	0	—	0	—	0	—	ns
t <sub>HZE</sub> <sup>(2)</sup>	$\overline{WE}$ LOW to High-Z Output	—	4	—	5	—	6	—	7	ns
t <sub>LZE</sub> <sup>(3)</sup>	$\overline{WE}$ HIGH to Low-Z Output	3	—	3	—	3	—	3	—	ns

# 如何确定CPU主频?



- ❖ 流水线要求每个机器周期完成一个步骤，包括存储器读/写
- ❖ 多周期CPU也大都要求在一个步骤内完成存储器的一次访问
- ❖ 完成一次存储访问的时间?
- ❖ 如何实现? 50MHz

# 串行接口芯片8251



串行接口，可用于同步或异步传送

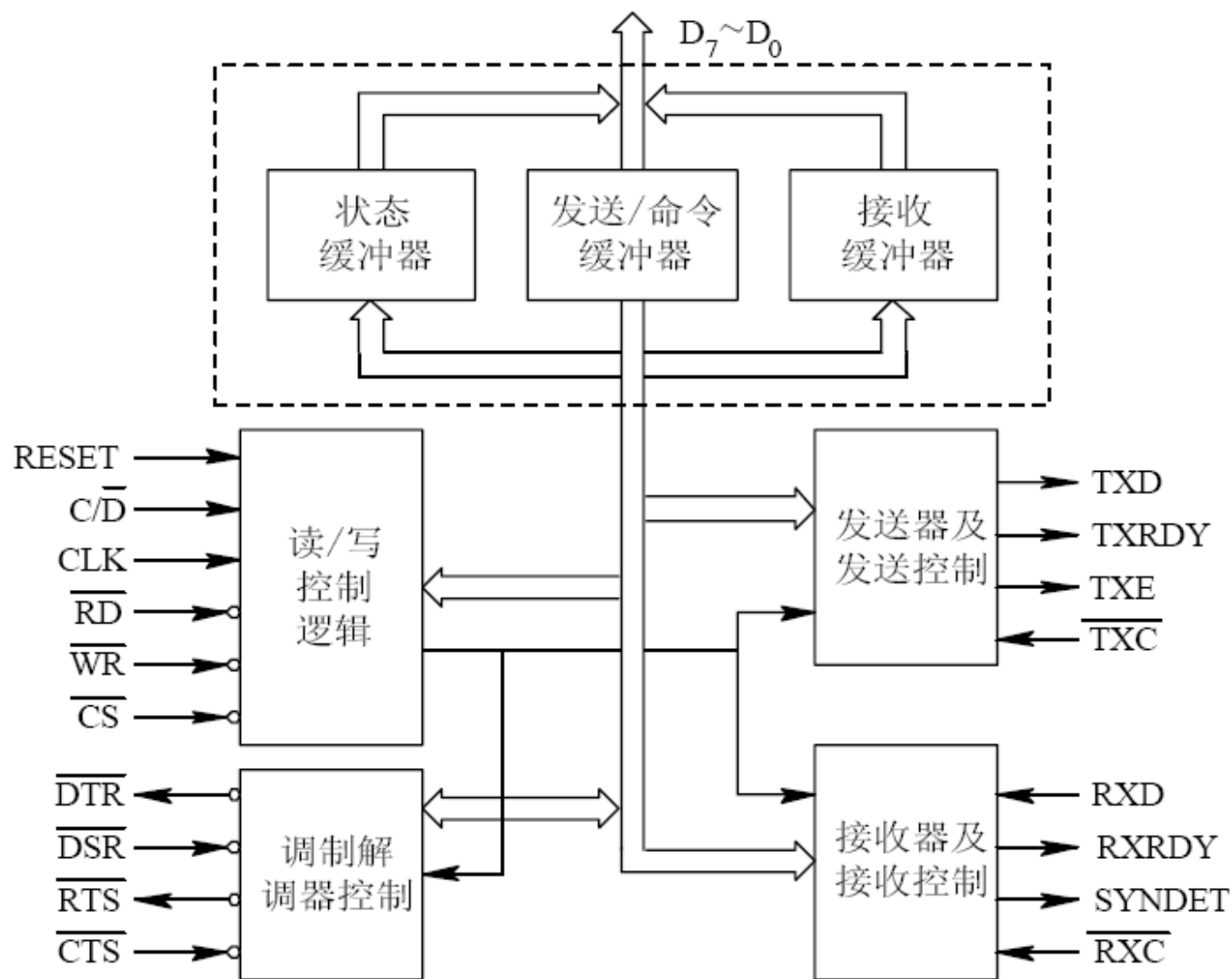
## 同步传送

- 5~8位/字
- 支持内同步或外同步
- 自动插入同步字符

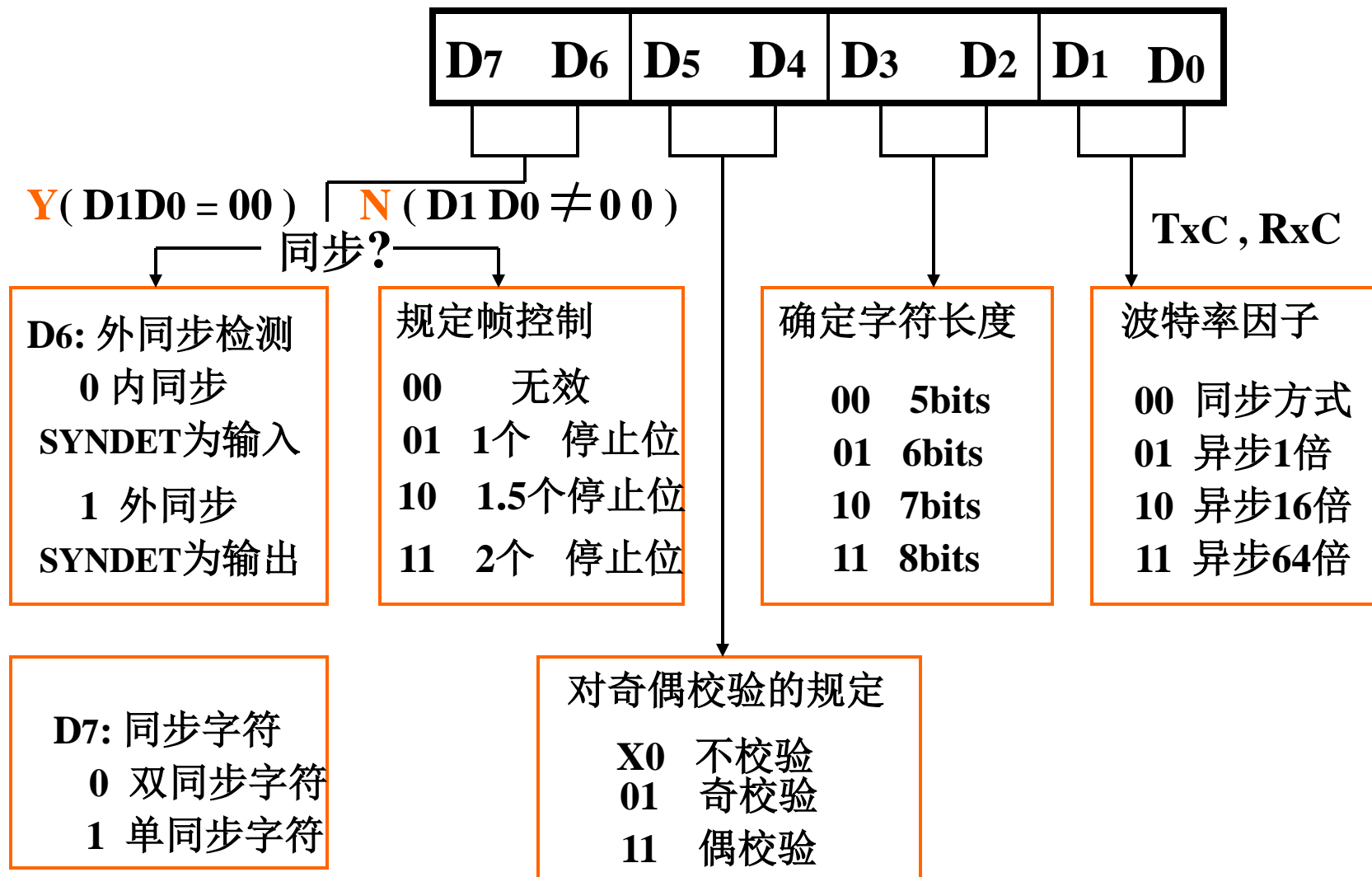
## 异步传送

- 5~8位/字
- 时钟：1、16或64倍波特率
- 停止位：1、1.5或2位
- 可检测假启动
- 全双工
- 双缓冲发送器和接受器
- 可检测奇偶错、数据丢失错和帧错

# 8251结构框图

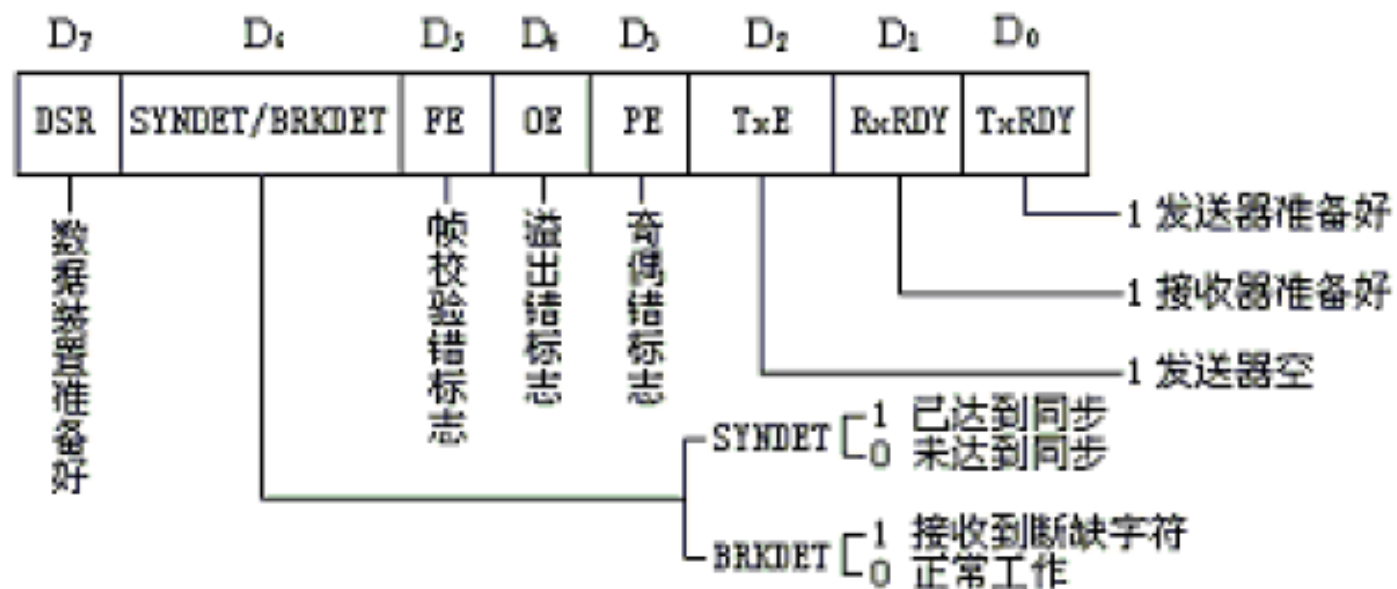


# 方式命令字的格式





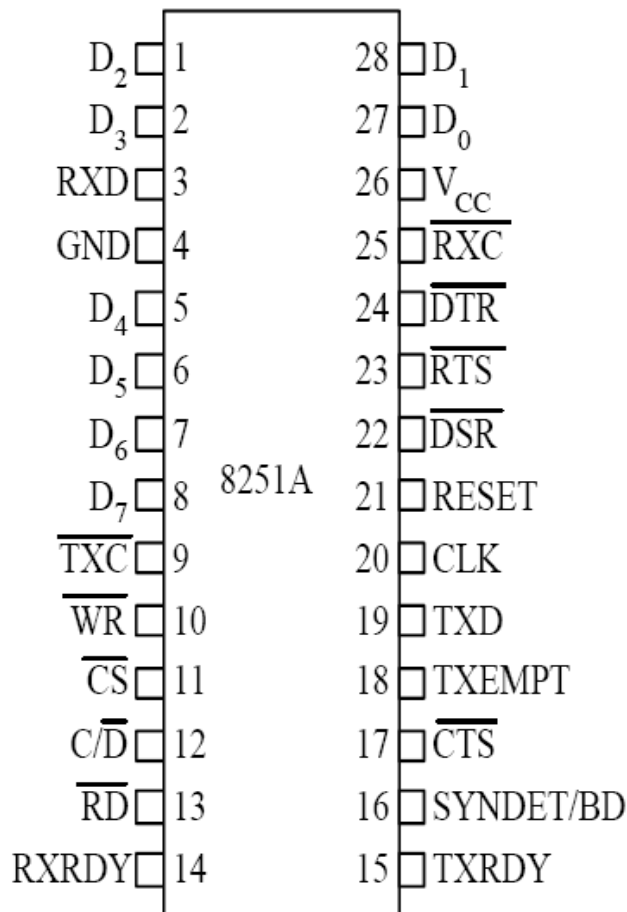
# 接口状态寄存器的内容格式



# INTEL 8251 串行接口芯片



清华大学  
Tsinghua University



器件引脚图

**D7~D0 : I/O数据**

**CLK: 主时钟**

**/RxC,RxD: 接收时钟、数据**

**/TxC,TxD: 发送时钟、数据**

**/CS: 片选信号**

**/WR、/RD: 写、读命令**

**C/  $\overline{D}$ : 控制 / 数据信号**

**RESET: 总清信号**

**RxRDY: 接收准备就绪**

**TxRDY: 发送准备就绪**

**TxEMPTY: 发送寄存器空**

**/DTR、/DSR:**

**/RTS、/CTS:**

# 工作命令字的格式



D7	D6	D5	D4	D3	D2	D1	D0
EH	IR	RTS	ER	SBBK	RxE	DTR	TxE <sub>N</sub>

## 接口状态寄存器的内容格式

D7	D6	D5	D4	D3	D2	D1	D0
DSR	SYNDET	FE	OE	PE	TxE	RxRDY	TxRDY

# TEC-2000 机串行口初始化的程序



```
MVRD R0, 4Eh
```

```
OUT 81h
```

```
MVRD R0, 37h
```

```
OUT 81h
```

方式设置: 0 1 0 0 1 1 1 0

1个停止位

无奇偶校验

16\*波特率

字符为 8 bits

命令设置: 0 0 1 1 0 1 1 1

不用

请求发送

错误标志复位

不用

允许接收

数据终端准备好

允许发送

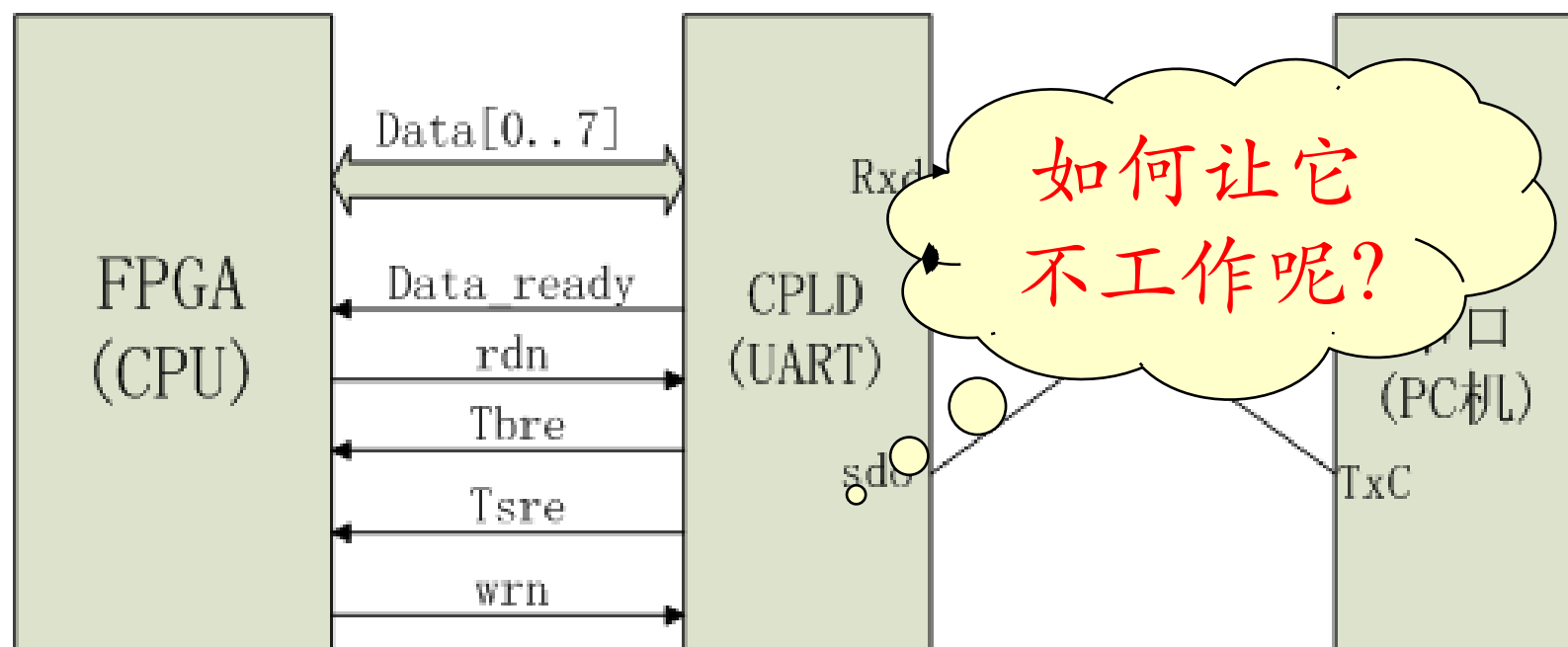
对 OUT 指令而言,

81h、91h 为控制寄存器地址;

对 IN 指令而言,

81h、91h 为状态寄存器地址;

# THINPAD串口连接方式



- ✿ rdn、Wrn控制读/写
- ✿ 读操作时，Data\_ready为1时，数据已经准备好
- ✿ 写操作时，Tbre且Tsre为1时，可进行写操作

# 监控程序中的写操作



TESTW:

NOP	MFPC R7
LI R6 0x00BF	ADDIU R7 0x0003
SLL R6 R6 0x0000	NOP
ADDIU R6 0x0001	B TESTW
LW R6 R0 0x0000	LI R6 0x00BF
LI R6 0x0001	SLL R6 R6 0x0000
AND R0 R6	LI R0 0x000A
BEQZ R0 TESTW	SW R6 R0 0x0000
;BF01&1=0 则等待	NOP
NOP	
JR R7	
NOP	

- ✚ 将BF01定向到UART
- ✚ 读BF01时，UART应该返回一个状态值
- ✚ 最低1位反映UART是否可写
- ✚ 若可写，则往BF00中写入字符

# 监控程序中的读操作



TESTR:

NOP

LI R6 0x00BF

SLL R6 R6 0x0000

ADDIU R6 0x0001

LW R6 R0 0x0000

LI R6 0x0002

AND R0 R6

BEQZ R0 TESTR

;BF01&2=0 则等待

NOP

JR R7

NOP

MFPC R7

ADDIU R7 0x0002

B TESTR

NOP

LI R6 0x00BF

SLL R6 R6 0x0000

LW R6 R5 0x0000

LI R6 0x00FF

AND R5 R6

NOP

- 将BF01定向到UART
- 读BF01时，UART应该返回一个状态值
- 次低1位反映UART是否可读
- 若可读，则从BF00中读入字符

# 对存储/外设的控制信号



## ✚ 信号源

- ✚ RAM1\_EN、RAM1\_OE、RAM1\_RW

- ✚ RAM2\_EN、RAM2\_OE、RAM2\_RW

## ✚ 对存储器的控制信号

- ✚ /CE /OE /WE

## ✚ 对串口UART的控制信号

- ✚ Rdn、Wrn

- ✚ Data\_ready、tbre、tsre



# 已有的基础



- ❖ 监控程序
- ❖ THCO MIPS指令系统
  - ❑ Project1
  - ❑ 思考和实践
- ❖ ALU及数据通路
  - ❑ Project2、思考和实践
- ❖ 串口、存储访问
  - ❑ Project3

# 基本步骤



## ✚ 确定目标

- ✚ 多周期/流水、扩展方案、展示方案
- ✚ 工作计划
- ✚ 组内分工

## ✚ 概要设计

- ✚ 数据通路设计：每条指令在数据通路上的实现过程
- ✚ 指令流程表：每条指令每个步骤所完成的功能
- ✚ 流水段设计
- ✚ 控制信号设计
- ✚ 模块设计（各模块基本功能和接口）

## ✚ 详细设计

- ✚ 各模块编码实现

## ✚ 编码实现

## ✚ 模拟分析

## ✚ 硬件调试

## ✚ 总结报告

# 我们还能做什么？



## 硬件

- ❑ 对流水线进行完善
  - ◆ 结构冲突、数据冲突、控制冲突
- ❑ 中断
- ❑ 分时操作
- ❑ 双机通信

## 软件

- ❑ 监控程序的完善
- ❑ 有创意的应用
- ❑ 性能分析和评价

# 时间安排



## ❖ 截止时间

- ❑ 实验检查：12月8日
- ❑ 实验报告：12月19日
- ❑ 实验答辩：1月6日

## ❖ 阶段检查

- ❑ 11月21日：确定目标和数据通路设计
- ❑ 11月25日：控制器设计
- ❑ 11月28日：初步代码

## ❖ 实验室开放及助教

- ❑ 工作日白天全时，晚上和周末预约
- ❑ 开放时间有助教辅导

# 实验成果



## ✦ 完整的实验报告

- ❑ 实验目标
- ❑ 实验内容
- ❑ 实验展示
- ❑ 实验心得体会

## ✦ 完整的源代码

- ❑ 设计框图
- ❑ 指令流程表
- ❑ VHDL代码
- ❑ 其他代码

# 下面三堂课的教学安排



- ✿ 分班上实验课，上课时间不变
- ✿ 上课地点变更为（以分组名单中的班号为准）
  - ✻ 计21： 五教 5101
  - ✻ 计22： 东主楼 9区419
  - ✻ 计23： 东主楼 9区422
  - ✻ 计24： 东主楼 9区423
  - ✻ 计25： 东主楼 9区223
- ✿ 要求
  - ✻ 按时上课，课前签到，出勤情况将影响实验成绩
  - ✻ 按时完成实验任务
  - ✻ 与软工联合小组自行安排

## ✚ 分析THCO MIPS指令系统

### ▣ 指令格式

◆ 操作码、操作数地址

### ▣ 寻址方式

◆ 寄存器寻址、立即数寻址、基址寻址

## ✚ 设计数据通路

## ✚ 划分指令执行步骤

◆ 多周期/指令流水