

实验四 模型机时序部件的实现

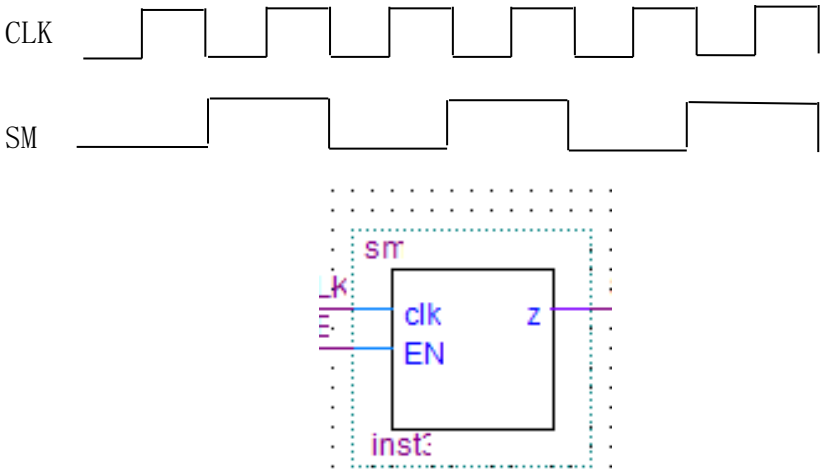
一、实验目的

- 1. 熟悉计数器、寄存器和 RAM 的工作原理。
- 2. 了解模型机中 SM 的作用。
- 3. 学会使用 VHDL 语言设计时序电路。

二、实验背景

1. SM

模型机中所有指令的执行都是一个周期完成取指令，一个周期执行指令。如何区分当前周期是取指令还是执行呢？这就需要SM。当SM=0时，该周期完成取指令，当SM=1时，该周期执行指令。SM的功能及接口如下：

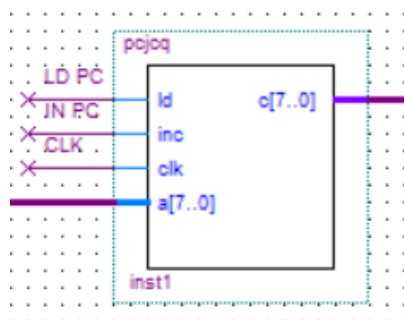


CLK	EN	功能
	1	$SM \leftrightarrow SM \text{取反}$
	0	SM不变

2. 指令计数器PC

指令计数器PC保存的是下一条指令在RAM中存放的地址。

CPU执行一条指令，根据PC中存放的指令地址，将指令从RAM读出写入指令寄存器IR中，此过程称为“取指令”，与此同时，PC中的地址自动加1，指向下一条指令在RAM中的存放地址。跳转指令如JMP、JZ、JC让程序跳转至指定地址去执行，这时PC需要装载跳转地址。因此，指令计数器PC的接口及功能如下：

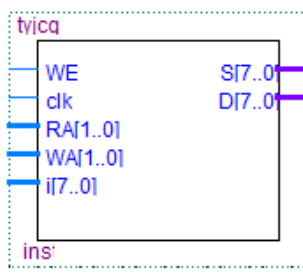


CLK	IN PC	LD PC	功能
	1	0	c[7..0]中数据自加1
	0	1	a[7..0] 向入c[7..0]

3. 通用寄存器组

寄存器是CPU内的重要组成部分，模型机的通用寄存器组包含3个8位寄存器

A、B、C，实现对此3个寄存器的读写操作。其接口及功能如下：



操作	CLK	WE	RAA[1..0]	RWBA[1..0]	功能
读			00或01或10	00或01或10	根据RAA[1..0]的值从A,B,C中选择一个寄存器的值由S口输出 根据RWBA[1..0]的值从A,B,C中选择一个寄存器的值由D口输出
写		0	XX	00或01或10	控制信号WE为0，根据RWBA[1..0]的值，在时钟下降沿将外部输入写入A,B,C三个寄存器中的某个寄存器。

4. RAM

半导体存储器的种类很多，从功能上可以分为只读存储器ROM和随机存储器RAM两大类。

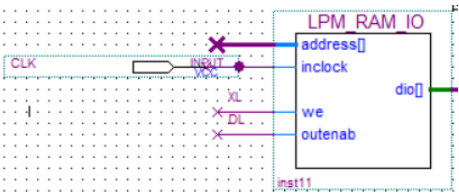
随机存储器RAM是与CPU直接交换数据的内部存储器，也叫主存（内存）。它

可以随时读写，而且速度很快，通常作为操作系统或其他正在运行中的程序的临时数据存储媒介。

存储元是构成存储器的存储介质，它可存储一个二进制位。由若干个存储元组成一个存储单元，然后再由许多存储单元组成一个存储器。一个存储器包含许多存储单元，每个存储单元可存放一个字节。每个存储单元的位置都有一个编号，即地址，一般用十六进制表示。一个存储器中所有存储单元可存放数据的总和称为它的存储容量。比如，一个存储器的地址码由8位二进制数（即2位十六进制数）组成，则可表示2的8次方，即256个存储单元地址，每个存储单元存放一个字节，则该存储器的存储容量为256×8，即2Kbit。

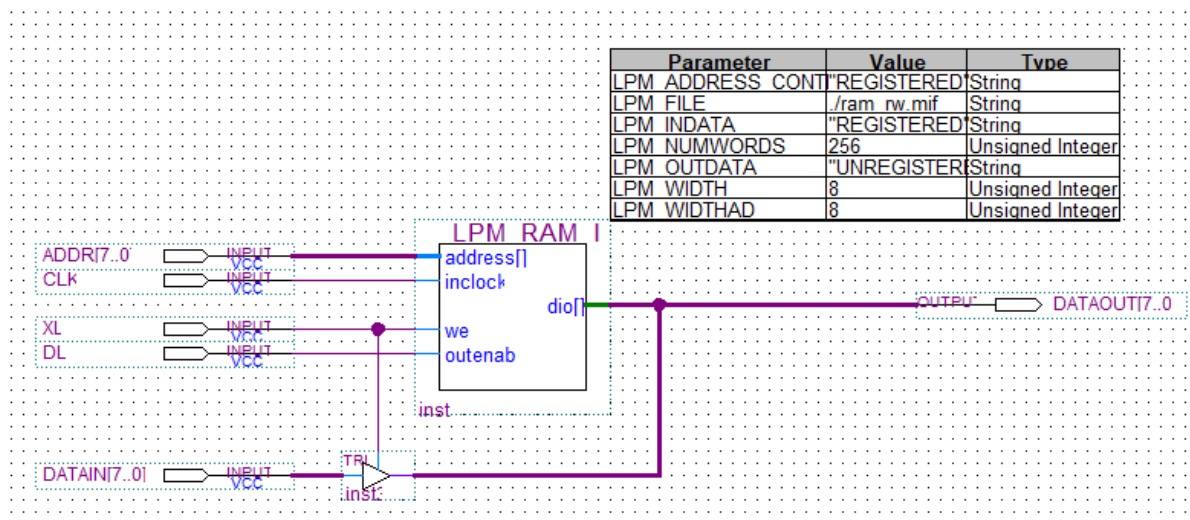
本实验可采用Quartus中已有的参数化模块来定制RAM功能，在【Symbol】元件库的【megafunctions】|【storage】中选择LPM_RAM_IO，创建RAM时加载初始化数据文件，初始化文件的创建过程是：【File】|【New】|【Memory Initialization File】。

LPM_RAM_IO的符号及功能如下：



CLK	We (XL)	outenab (DL)	功能
	0	0	Dio<=高阻态Z
	1	0	Dio的数据写入address所指定的存储单元
	0	1	address所指定的存储单元数据从dio输出

ADDR[7..0]指定访问RAM的地址，加载时钟CLK，XL为1，将外部输入DATAIN[7..0]写入RAM的对应存储单元。不改变ADDR[7..0]的值，这时DL为1，读取RAM，查看DATAOUT[7..0]中的输出是否跟前面写入的数据是否一致，从而学习对RAM的读写操作。RAM读写操作的参考电路如图所示：



三、实验内容

1. 用 VHDL 语言设计 SM;
2. 用 VHDL 语言设计一个 8 位的指令计数器 PC;
3. 用 VHDL 语言设计 3 个 8 位寄存器组成的寄存器组，实现读写操作。
4. 用 LPM_RAM_IO 定制一个 256*8 的 RAM，实现对 RAM 的读写操作;

四、实验要求

1. 进实验室前，请写一份预习报告；如有疑问，可在学习通平台相互讨论。
2. 预习报告内容有:

- 用 VHDL 语言编写 SM;
- 用 VHDL 语言编写一个 8 位的程序计数器 PC ;
- 用 VHDL 语言编写 3 个 8 位寄存器组成的寄存器组，实现读写操作。
- 采用 LPM_RAM_IO 定制一个 256*8 的 RAM，实现对 RAM 的读写操作。

3. 在文本编辑区使用 VHDL 硬件描述语言设计逻辑电路，再利用波形编辑区进行仿真，以此验证电路的逻辑功能是否正确，最后在 Tool 下用 netlist viewer 查看 RTL viewer，以查看实现的 RTL 电路图。

5. 实验结束前，由指导老师检查了仿真波形后方可离开。

6、最后撰写实验报告，提交到学习通平台，并在平台上分享设计的警告、资源消耗以及 RTL 视图。

7、评判各种实现方案，并打分。

五、思考题

1. 时钟周期的上升沿实现对 RAM 的读写操作，为何 PC、SM 以及寄存器组的操作是下降沿完成？

2. 总结 VHDL 语言描述时序部件的方法和常用语句。