

实验四 模型机时序部件的实现

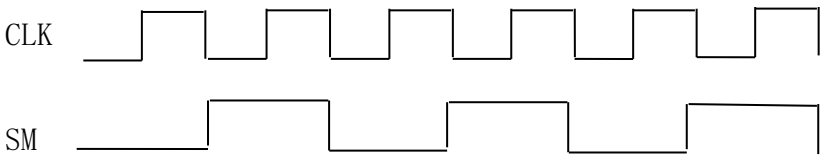
一、实验目的

- 1. 了解模型机中 SM 的作用。
- 2. 熟悉指令寄存器、状态寄存器、指令计数器、寄存器的工作原理
- 3. 学会使用 VERILOG 语言设计时序电路。

二、实验背景

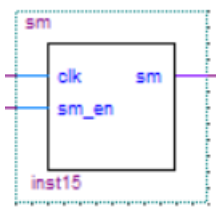
1. SM

模型机中所有指令都要求两个周期完成，其中一个周期读取指令，一个周期执行指令。如何区分当前周期是取指令还是执行指令呢？这就需要SM配合。SM为0是读取指令周期；SM为1是执行指令周期。SM的功能及封装如下：



SM的功能表：


CLK	SM_EN	功能	备注
	1	$SM \leftrightarrow SM取反$	SM初始值为0

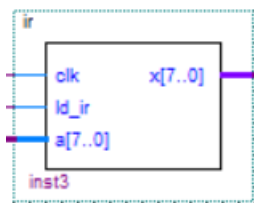


2、指令寄存器IR

指令寄存器（IR，Instruction Register）用于暂存当前要执行的指令。指令寄存器将总线送来的指令存入8位寄存器中，但并不是每次总线上的数据都需要寄存，因为总线上有时传输指令，有时传输数据。当控制信号LD_IR为1时，指令寄存器在时钟信号CLK的下降沿将总线传输的指令写入寄存器。

指令寄存器IR是一个8位寄存器。其功能及封装如下：


CLK	Ld_ir	功能	备注
	1	a[7:0]写入x[7:0]	X初始值00000000

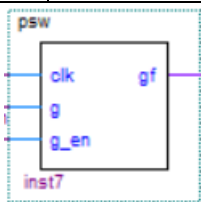


3、状态寄存器PSW

状态寄存器PSW是计算机系统的核心部件——运算器的扩展部分。本模型机PSW用来存放SUB指令执行结果的状态标志，如果 $R_d > R_s$ ，则 $G=1$ ，否则 $G=0$ 。有些机器也将PSW称为标志寄存器FR（Flag Register）。

本模型机PSW是一个1位寄存器。其功能及封装如下：

CLK	g_en	功能	备注
	1	g写入gf	gf初始值为0





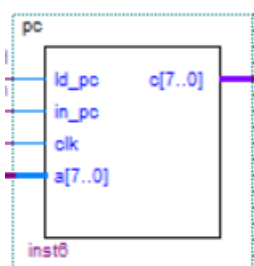
4. 指令计数器PC

指令计数器PC存储当前指令在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“读取指令”。在每条指令读取后，指令计数器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、JG让程序跳转至指定地址去执行，这时PC需要装载跳转地址。

模型机的指令计数器PC是一个8位计数器，其的功能及封装如下：


CLK	in_pc	ld_pc	功能	备注
	1	0	c[7..0]中数据自加1	PC初始值为00000000 (第一条指令在RAM中的存放地址)
	0	1	a[7..0]写入c[7..0]	

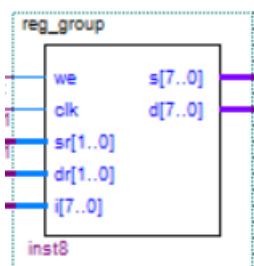


5. 通用寄存器组

寄存器主要用来保存操作数和运算结果等信息，从而节省从RAM中读取操作数所需占用总线和访问存储器的时间。

模型机的通用寄存器组包含4个8位寄存器R0、R1、R2、R3，可对这4个寄存器进行读写操作。其功能及封装如下：

操作	CLK	WE	功能
读			根据SR[1..0]的值从R0、R1、R2、R3中选择一个寄存器的值从S口输出 根据DR[1..0]的值从R0、R1、R2、R3中选择一个寄存器的值从D口输出
写		1	控制信号WE为1，根据DR[1..0]的值，在CLK下降沿将外部输入i写入R0、R1、R2、R3中的一个寄存器。



6. 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。

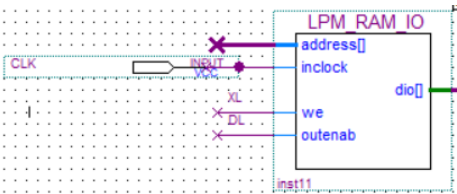
随机存储器RAM是与CPU直接交换数据的内部存储器，也叫主存（内存）。它可以随时读写，而且速度很快，通常作为操作系统或其他正在运行中的程序的临时数据存储媒介。

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 2^8 次方，即256个存储单元地址，每个存储单元存放一个字节，则该存储器的存储位数为 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】。

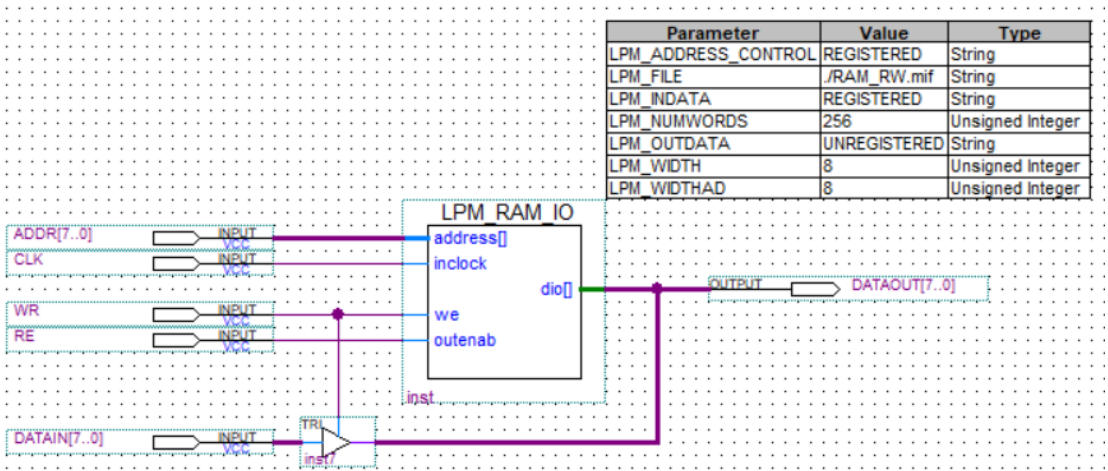
LPM_RAM_IO的符号及功能如下:



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

对LPM_RAM_IO进行读写操作:

LPM_RAM_IO的读写操作参考电路如下图所示。ADDR[7..0]指定读写的存储单元地址，首先让RE为0，WR为1，时钟CLK上升沿，将外部输入DATAIN[7..0]写入RAM的对应存储单元。ADDR[7..0]的值保持不变，接着让RE为1，WR为0，时钟CLK上升沿，读取同一存储单元数据输出至DATAOUT[7..0]，比较此时DATAOUT[7..0]的输出与DATAIN[7..0]写入的数据是否一致，从而验证对RAM同一存储单元进行读写的正确性。



三、实验内容

1. 用 VERILOG 语言设计 SM;
2. 用 VERILOG 语言设计一个 8 位的指令寄存器 IR;

3. 用 VERILOG 语言设计一个 1 位的状态寄存器 PSW;
4. 用 VERILOG 语言设计一个 8 位的指令计数器 PC;
5. 用 VERILOG 语言设计 4 个 8 位寄存器组成的寄存器组, 实现读写操作;
6. 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. 完成学习通平台的实验作业。
2. 采用 VERILOG 语言设计逻辑电路, 再利用波形编辑区进行仿真验证, 以此验证电路的正确性。
3. 在 Tool 下用 netlist viewer 查看 RTL viewer, 了解语句描述对应的 RTL 视图。
4. 实验结束前, 由指导老师检查了仿真波形后方可离开。
5. 最后撰写实验报告, 提交至学习通平台, 做得好的同学将在学习通平台分享设计。

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、IR、PSW 以及寄存器组的操作是下降沿完成?
2. 采用 VERILOG 语言描述时序部件应该采用阻塞赋值语句还是非阻塞赋值语句?
3. 通用寄存器组只有 WE 的控制信号, 实现通用寄存器组读操作的电路是组合电路还是时序电路? 请大致画出对寄存器组进行读操作的电路部分。