实验3 Cache 控制器设计

1．实验目的

(1) 掌握Cache 控制器的原理及其设计方法。

(2) 熟悉FPGA应用设计及EDA 软件的使用。

(3) 熟悉Vivado软件的使用及FPGA应用设计。

2．实验原理

Cache是介于CPU和主存之间的小容量存储器，包括管理在内的全部功能由硬件实现，对程序员是透明的，在一定程度上解决了CPU和主存之间的速度差异。与主存容量相比，Cache的容量很小，它保存的内容只是内存内容的一个子集，且Cache与主存的数据交互以块为单位。把主存中的块放到Cache中时必须把主存地址映射到Cache中，即确定位置的对应关系，并采用硬件实现，以便CPU给出的访存地址能够自动变换成Cache地址。由于程序访问的局部性，使得主存的平均读出时间接近Cache的读出时间，大大提高了CPU的访存效率。

地址映射方式有全相联方式、直接相联方式、组相联方式，本实验采用的是直接方式，这种变换方式简单而直接，硬件实现很简单，访问速度也比较快，但是块的冲突率比较高。其主要原则是：主存中一块只能映象到Cache的一个特定的块中。

假设主存的块号为B，Cache的块号为b，则它们之间的映象关系可以表示为：b = B mod Cb其中，Cb是Cache的块容量。设主存的块容量为Mb，区容量为Me，则直接映象方法的关系如图3.19所示。把主存按Cache的大小分成区，一般主存容量为Cache容量的整数倍，主存每一个分区内的块数与Cache的总块数相等。直接映象方式只能把主存各个区中相对块号相同的那些块映象到Cache中同一块号的那个特定块中。例如，主存的块0只能映象到Cache 的块0中，主存的块1只能映象到Cache的块1中，同样，主存区1中的块Cb（在区1中的相对块号是0）也只能映象到Cache 的块0中。根据上面给出的地址映象规则，整个Cache 地址与主存地址的低位部分是完全相同的。



图1直接相联映像方式

直接映象方式的地址变换过程如图1所示，主存地址中的块号B与Cache地址中的块号b 是完全相同的。同样，主存地址中的块内地址W与Cache地址中的块内地址w也是完全相同的，主存地址比Cache地址长出来的部分称为区号E。



图2直接相联地址变换

在程序执行过程中，当要访问Cache时，为了实现主存块号到Cache块号的变换，需要有一个存放主存区号的小容量存储器，这个存储器的容量与Cache的块数相等，字长为主存地址中区号E 的长度，另外再加一个有效位。

在主存地址到Cache地址的变换过程中，首先用主存地址中的块号去访问区号存储器（按地址访问）。把读出来的区号与主存地址中的区号E进行比较，根据比较结果和与区号在同一存储字中的有效位情况作出处理。如果区号比较结果相等，有效位为‘1’，则Cache 命中，表示要访问的那一块已经装入到Cache中了，这时Cache地址（与主存地址的低位部分完全相同）是正确的。用这个Cache地址去访问Cache，把读出来的数据送往CPU。其他情况均为Cache没有命中，或称为Cache失效，表示要访问的那个块还没有装入到Cache中，这时，要用主存地址去访问主存储器，把读出来的一个字送往CPU，同时，把包括被访问字在内的一块都从主存储器中读出来，送往Cache的相应块。

Cache和CPU以及存储器的连接如图3所示，本实验要在FPGA中实现Cache及其地址变换逻辑（也叫Cache控制器），采用直接相联地址变换，只考虑CPU从Cache读数据，不考虑CPU写回数据的情况。

Cache控制器顶层模块如图4所示，主存地址为A7…A0,共8位，区号E取3位，这样Cache地址还剩5位，所以Cache容量为32个单元，块号B取3位，那么Cache分为8块，块内地址W取2位，则每块为4个单元。图4中，WCT为写Cache块表信号，CLR为系统总清零信号，A7…A0为CPU访问内存的地址，M为Cache失效信号，CA4…CA0为Cache地址，MD7…MD0 为主存送Cache的数据，D7…D0为Cache送CPU数据，T2为系统时钟，RD为CPU访问内存读信号，LA1和LA0为块内地址。



图3Cache 系统图



图4 Cache 控制器顶层模块图

3．实验内容

(1)在Vivado软件中，使用D触发器作为存储一个二进制代码的存储单元，设计一个8位的存储单元模块，按照基于硬件描述语言的设计流程输入该电路，进行编译、仿真验证后生成存储单元模块供上层设计调用。

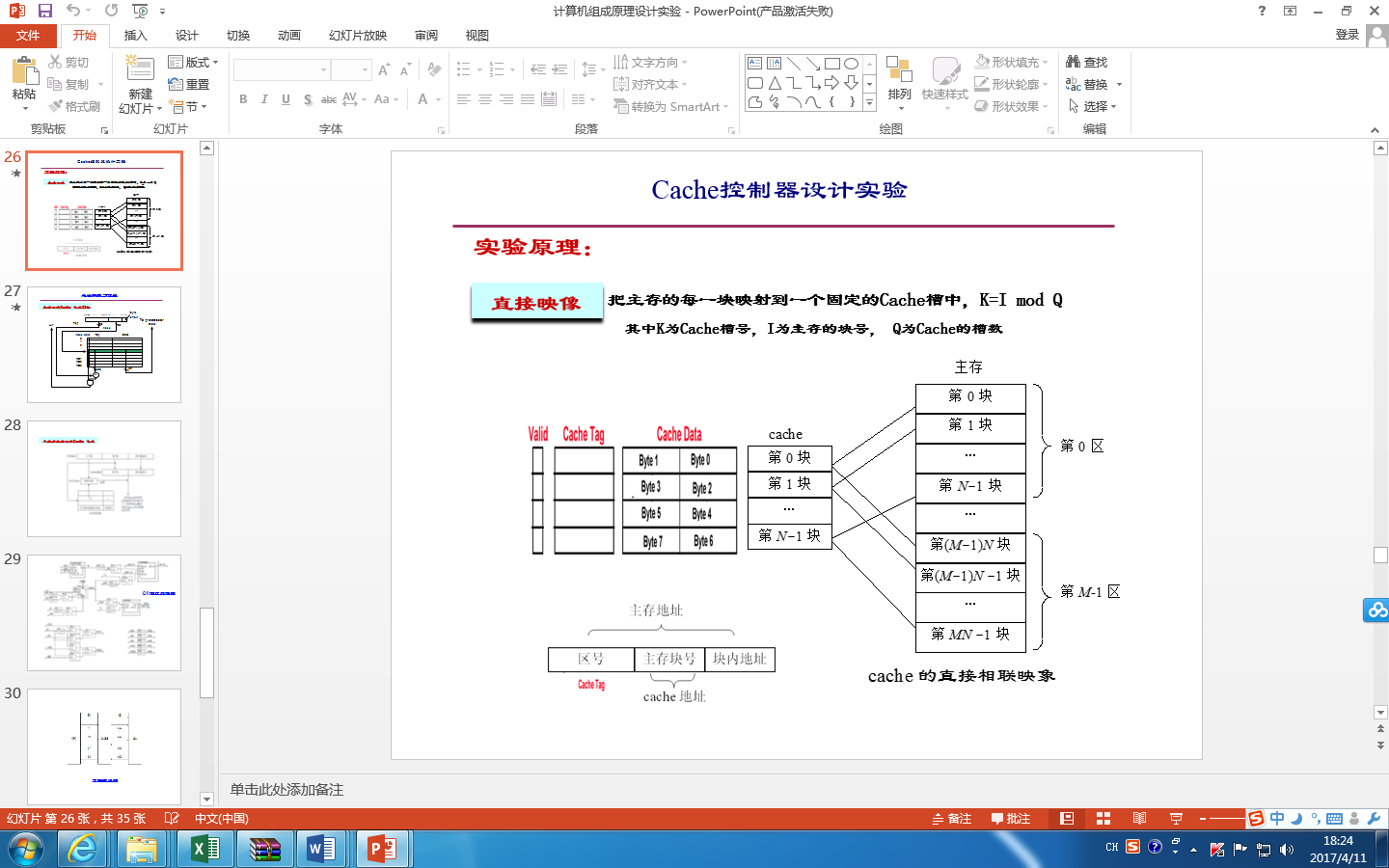
(2)根据对实验原理的描述，使用这个8位的存储单元构成一个32\*8位的Cache存储体，并在Vivado软件中实现该电路，进行编译、仿真验证后生成Cache存储体模块供上层调用。

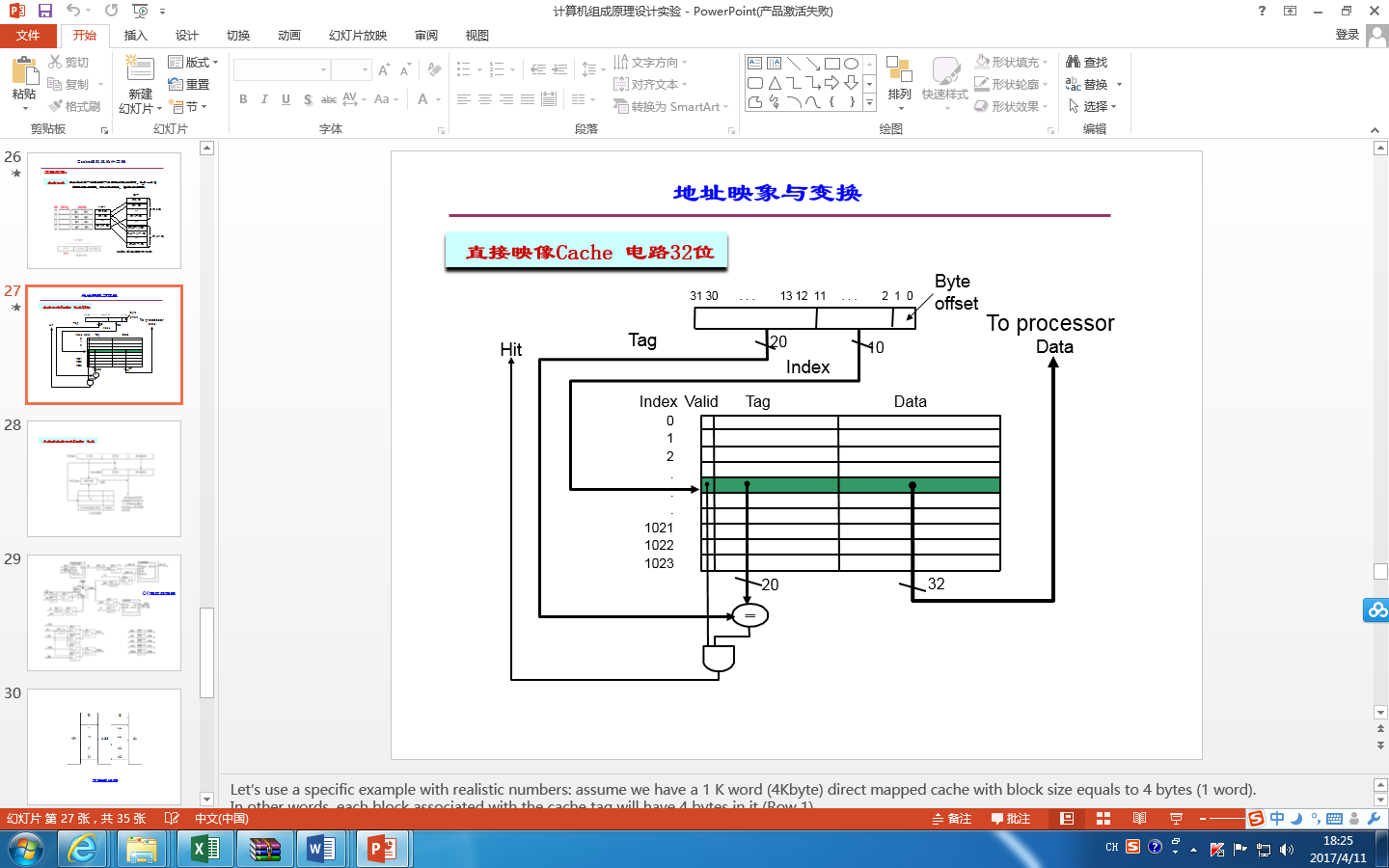
(3)参照上面两步，在Vivado软件中实现一个4位的存储单元模块；然后使用这个４位存储单元构成一个8\*4位的区表存储器，用来存放区号和有效位。由图2可知，还需要实现一个区号比较器，如果主存地址的区号Ｅ和区表中相应单元中的区号相等，且有效位为１，则Cache命中，否则Cache失效，标志位为M，M为0时表示Cache失效。最后，编译、仿真验证后生成区表存储器模块供上层调用。

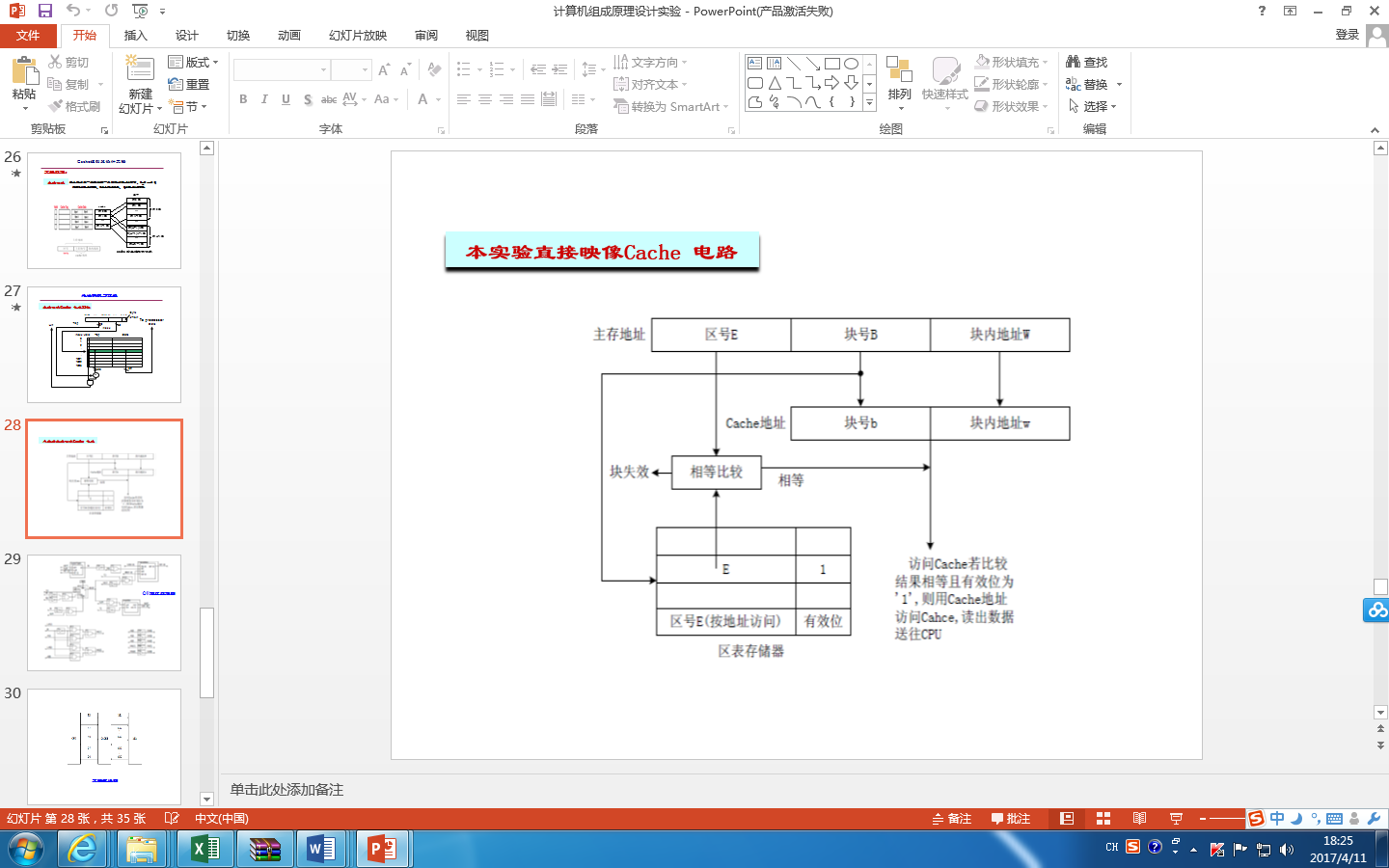
(4)当Cache命中时，就将Cache存储体中相应单元的数据送往CPU，这个过程比较简单；当Cache失效时，需要将主存中相应块中的数据读出写入Cache中，这样Cache控制器就要产生访问主存储器的地址和主存储器的读信号，由于每块占4个单元，所以需要连续访问4次主存，这就需要一个低地址发生器。在Vivado软件中，按照基于VerilogHDL的设计流程输入一个2位计数器，用于产生低2位地址00~11，进行编译、仿真验证后生成低地址发生器模块供上层调用。

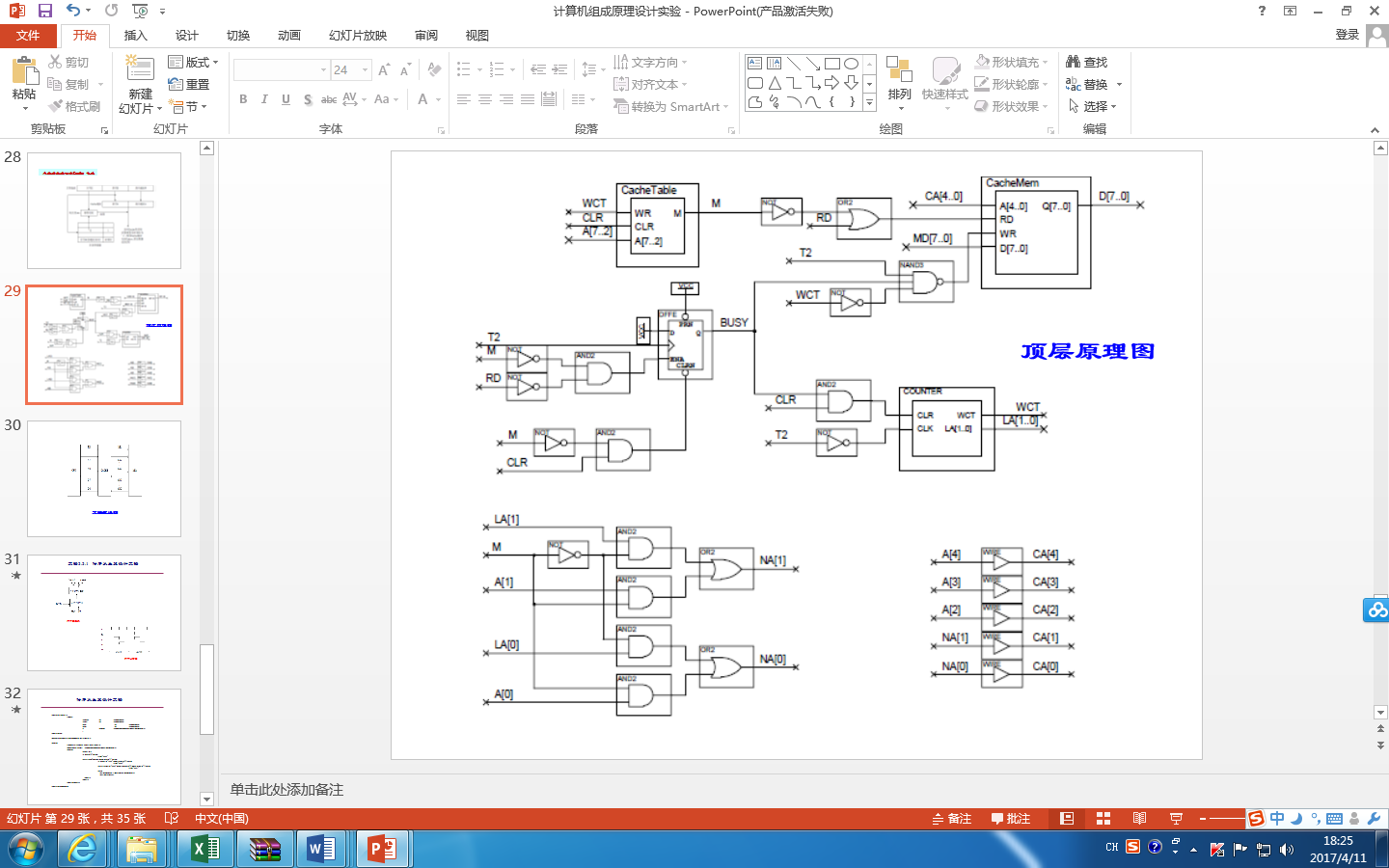
(5)将产生的低2位地址和CPU给出的高6位地址组合起来，形成访问主存储器的地址，M可以作为主存的读信号。这样在时钟T2的控制下，就可以将主存中相应的块写入Cache的相应块中，最后再修改区表。在Vivado软件中，参照图4给出的Cache存储器顶层模块，按照基于VerilogHDL的设计流程输入该电路，进行编译、仿真验证后生成编程文件。

思考题：对比分析全（组）相联映射与直接映射方式的区别。如何在Vivado 软件中实现全（组）相联映射方式的Cache存储器？

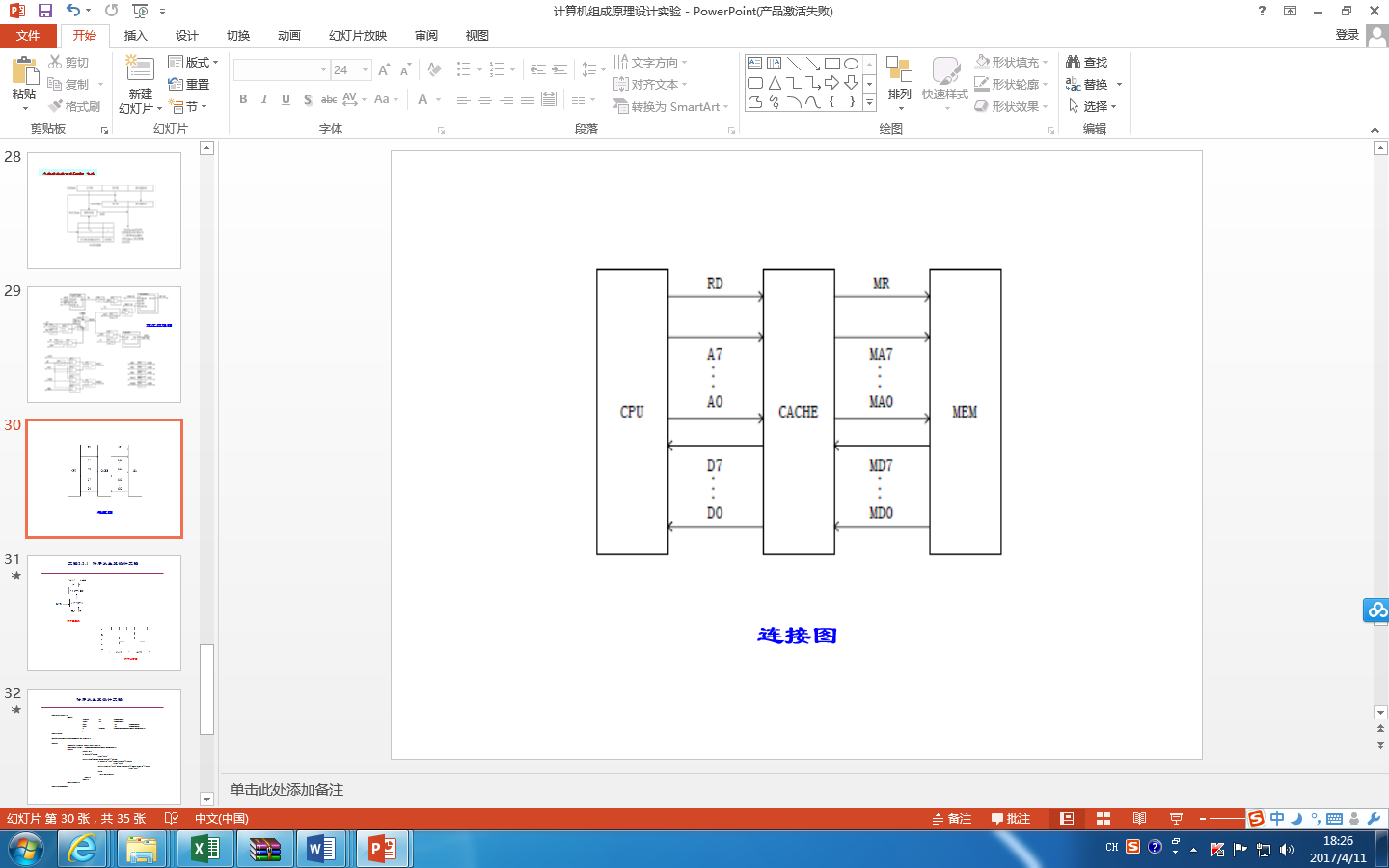








顶层模块图



CPU 读内存时 和 CACHE MEM之间的连接信号