

存储器实验

实验目的

- (1) 掌握随机存储器原理，学会 FPGA 内部存储器控制器的设计方法。
- (2) 掌握单端口与双端口 RAM（随机存储器）设计与实现。
- (3) 掌握 FIFO（先入先出存储队列）的设计与实现。

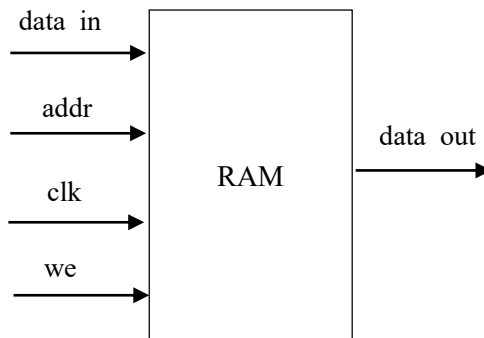
实验内容

- 1) 利用开发板片内存储器单元实现单端口 RAM 设计（带异步读和同步读两种模式）。
- 2) 实现双端口（同步与异步）RAM 设计。
- 3) 实现 FIFO 设计，FIFO 由存储单元队列或阵列构成，FIFO 没有地址，第一个被写入队列的数据也是第一个从队列中读出的数据。

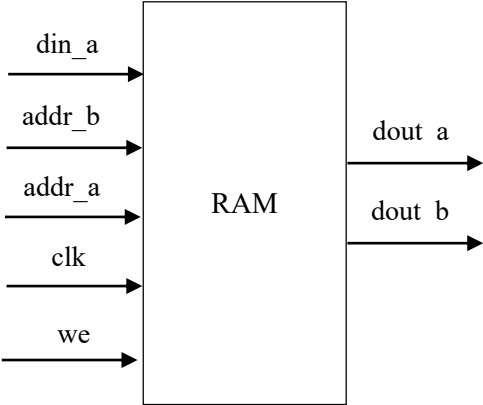
实验原理

RAM (random access memory) 又称“随机存储器”，存储单元的内容按需要随意取出或者存入，速度很快，但断电时将丢失数据，所以一般被作为临时数据的存储媒介。Basys3 开发板上拥有 1,800 Kbits 快速 RAM 块，可以根据需求定制 ROM、RAM 或者 FIFO。

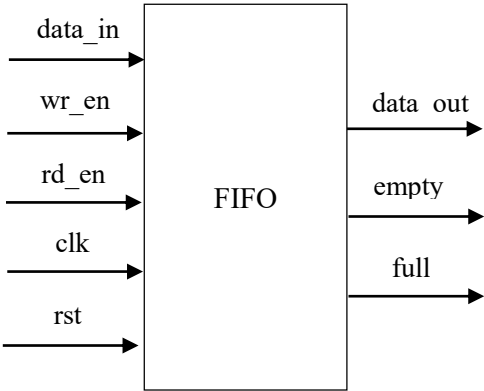
单端口 RAM 设计（带异步读和同步读两种模式），在时钟 (clk) 上升沿，采集地址 (addr)、输入数据 (data_in)、执行相关控制信息。当写使能 (we) 有效，则执行写操作，否则执行读取操作。同步与异步设计仅针对读操作：对于异步 RAM 而言，读操作为异步，即地址信号有效时，控制器直接读取 RAM 阵列；对于同步 RAM 而言，地址信号在时钟上升沿被采集，并保存在寄存器中，然后使用该地址信号读取 RAM 阵列，单端口 RAM 框图如下：



双端口（同步与异步）RAM，相对于单端口 RAM 而言，双端口 RAM 存在两个存取端口，并且可独立进行读写操作，具有自己的地址（addr_a、addr_b）、数据输入（din_a、din_b）/输出端口（dout_a、dout_b）以及控制信号。双端口 RAM 常用于视频/图像处理设计中。双端口 RAM 框图如下：



FIFO 是一个先入先出的存储队列，和 RAM 不同的是 FIFO 没有地址，第一个被写入队列的数据也是第一个从队列中读出的数据。FIFO 可以在输入输出速率不匹配时，作为临时存储单元；可用于不同时钟域中间的同步；输入数据路径和输出数据路径之间数据宽度不匹配时，可用于数据宽度调整电路。FIFO 的框图和信号功能如下：



信号	功能
data_in	数据输入
data_out	数据输出

wr_en	写使能
rd_en	读使能
clk	FIFO 时钟
rst	FIFO reset
empty	表示 FIFO 空
full	表示 FIFO 满

实验步骤

- 1) 在 Vivado 中新建项目，设计实现带异步读模式的单端口 RAM，在时钟（clk）上升沿，采集地址（addr）、输入数据（data_in）、执行相关控制信息。当写使能（we）有效，则执行写操作，否则执行读取操作。读操作为异步，即地址信号有效时，控制器直接读取 RAM 阵列。
- 2) 编写仿真文件，观察波形图，验证其正确性，并下载到开发板进行验证。
- 3) 在 Vivado 中新建项目，设计实现带同步读模式的单端口 RAM，与异步读模式不同的是，地址信号在时钟上升沿被采集，并保存在寄存器中，然后使用该地址信号读取 RAM 阵列。
- 4) 编写仿真文件，观察波形图，验证其正确性，并下载到开发板进行验证。
- 5) 在 Vivado 中新建项目，实现双端口（同步或异步）RAM 设计，相对于单端口 RAM 而言，双端口 RAM 存在两个存取端口，并且可独立进行读写操作，具有自己的地址（addr_a、addr_b）、数据输入（din_a、din_b）/输出端口（dout_a、dout_b）。
- 6) 通过仿真、下板验证其正确性。
- 7) 在 Vivado 中新建项目，实现 FIFO 设计，FIFO 由存储单元队列或阵列构成，和 RAM 不同的是 FIFO 没有地址，第一个被写入队列的数据也是第一个从队列中读出的数据，FIFO 有表示“空”和“满”的状态输出信号。
- 8) 通过仿真、下板验证其正确性。