Chapter 3 Sequential Logic

- *It's a poor sort of memory that only works backward
- 记忆如此悲怜，只能回溯过去
- Introduction
  - chap1，2中忽略了时间概念，只是输入输出的逻辑映射
    - 组合芯片：输出结果仅依赖输入变量
    - 无法维持自身状态
  - pc需要存取数据，所以要配备记忆单元
    - 记忆单元由时序芯片组成
- Background
  - 物理世界的时间为矢量，在计算机中离散化
  - 时钟Clock：
    - 时间的流逝由主时钟表示，提供连续的交变信号序列（基于振荡器）
    - 在0-1（低高电平）的交替变化形成周期，每一个周期是一个数字整数时间单位
    - 这个信号被同时传送到pc平台的每一个时序芯片中
    - 实际情况下会产生一定延迟，而在整数情况下不考虑（只要所有硬件稳定且给出正确输出）
    - 顺序逻辑
      - 提供了t-1的信息，计算t信息
      - t的输出取决于t-1的输入
      - 组合逻辑中，out（t）=f（in（t））
      - 顺序逻辑中，out（t）=f（in（t-1））：新旧信息不混杂
  - 触发器Flip-Flops
    - 需要移动信息（t-1到t）
    - 计算机最基本的时序单元为触发器，本书中使用数据触发器DFF
    - 1-bit in,1-bit out+一个时钟输入，根据主时钟信号交变
    - **DFF将前一个时间周期的输入值作为当前周期的输出**
    - 本课程中DFF为基本元件
  - 寄存器Registers
    - 具有记忆功能的设备，实现经典存储行out(t)=out(t-1)

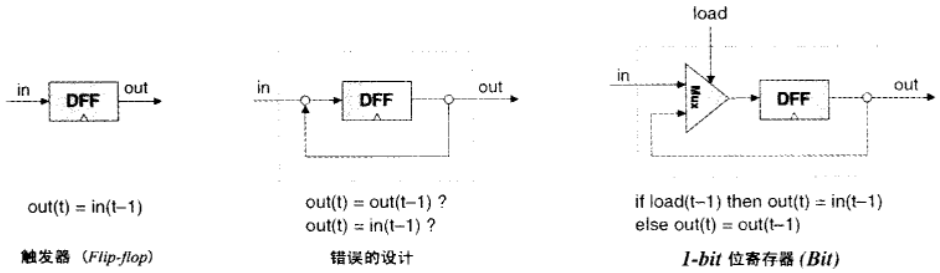

图 3.1　从 DFF 触发器到 1-bit 位寄存器。图中的小三角代表时钟信号的输入。用灰色小方块代表的芯片，以及用虚线框封装的整个芯片组都与时间存在相关性

- 通过Mux实现，Mux的选择位作为寄存器的load bit
  - 当load=1时，寄存器开始存储一个新值
  - 当load=0时，寄存器始终不变
- 多bit寄存器同理实现，基本设计参数为宽度（保存bit位的数量）容量用word表示

- 内存Memories
  - 很多寄存器堆叠起来，形成随机存取RAM单元
  - RAM：随机访问被选择的字，与访问顺序无关！O（1）：完全相同的时间轴

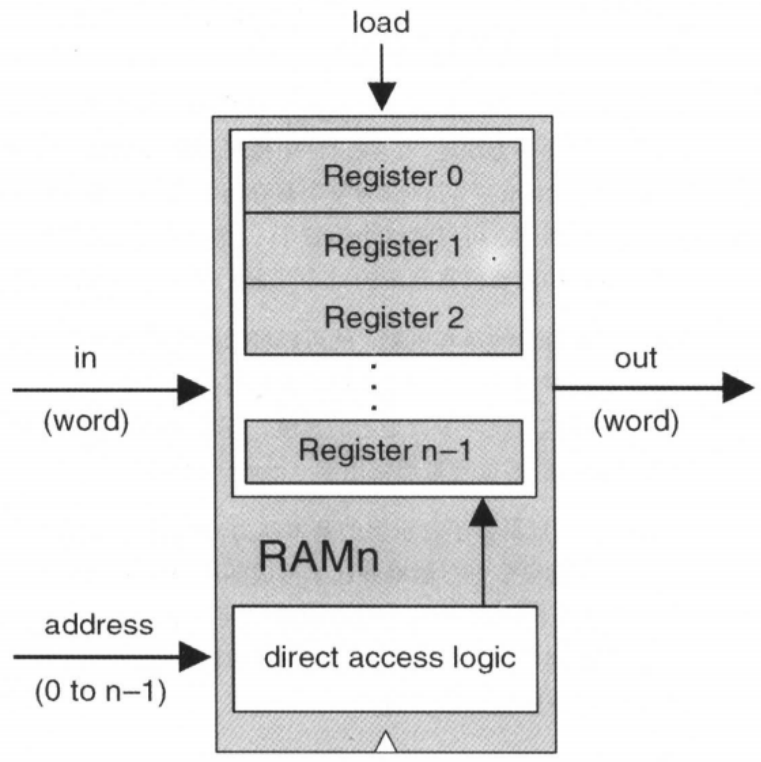

图 3.3　（概念上的）内存（RAM）芯片。RAM 的存储宽度和容量不定

  - 每一个俱存其都有一个地址(0base)
    - 输入k（logn位）来找到地址
  - Read logic
    - 首先确定选择地址

        - 从RAM中探查出对应寄存器
        - 会输出该寄存器的结果
    - Write logic
        - 确定地址
        - 设置in，load bit
        - 运行
- 计数器Counter
    - 一种时序芯片
    - 状态为整数，每经过一个时间周期，该整数增加1
- 时间问题
    - 嵌入DFF的芯片即为时序芯片
    - 被DFF赋予了维持状态（如memory）或对状态操作（如计数器）的能力
    - 由于DFF，输出变化仅发生在时钟周期的转换点上
    - 时序芯片可以进行离散化过程：作用——整个计算机系统的同步
        - 这将一组相互独立的硬件组件同步为协调统一的系统

- Specifiction
    - D触发器
        - 所有DFF连接同一个主时钟
        - 在每个时钟周期的起始点，所有DFF的输出被赋予了上个时钟周期的输入
        - 所以只要设计的芯片中含有DFF门，芯片都将继承对时间的相关性
        - DFF的实现使用反馈回路（也仅基于Nand）
    - 寄存器
        - 1bit位寄存器可以称为比特
        - 读写逻辑
    - 存储
        - 读写逻辑
    - 计数器
        - 与寄存器API类似，增加两个附加控制位reset，inc
        - load=1时，同寄存器
        - inc=1时，每个时钟周期自加
        - reset=1时，重设计数器数值为0
        - 如果什么都没有，则输出不变
- Implementation
    - 现代计算机并不总由标准触发器构建而成，性价比问题

以上内容整理于 幕布文档