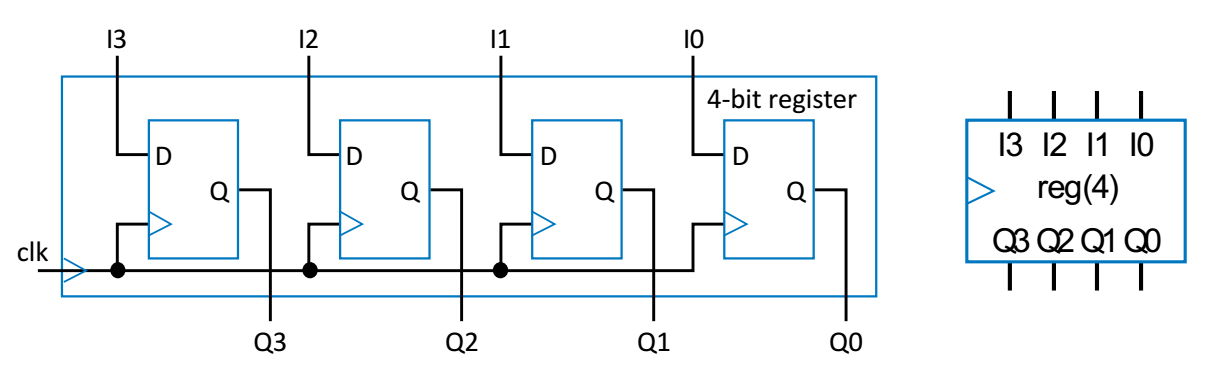
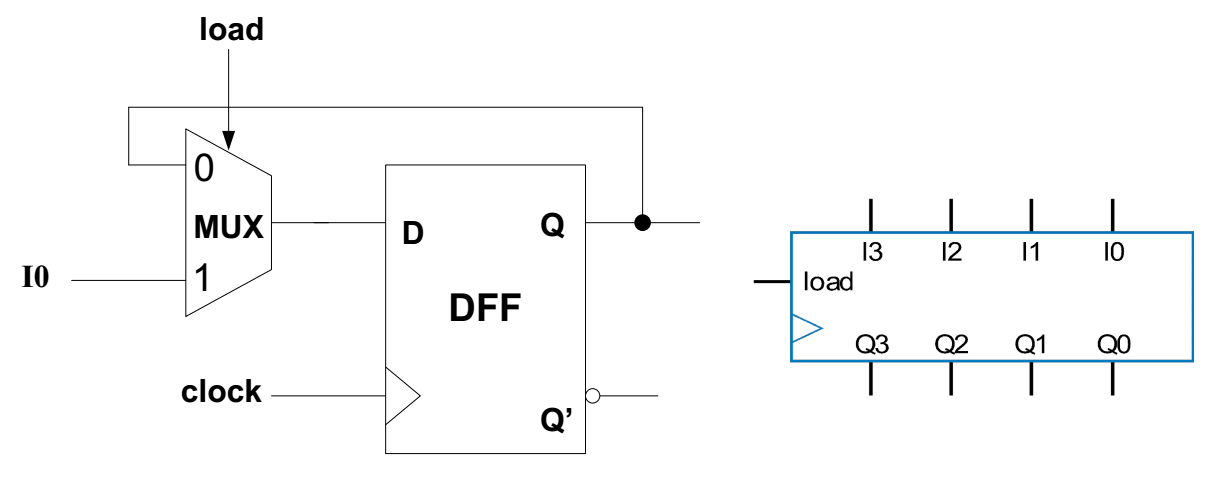
**Chapter 9 Register and shifter**

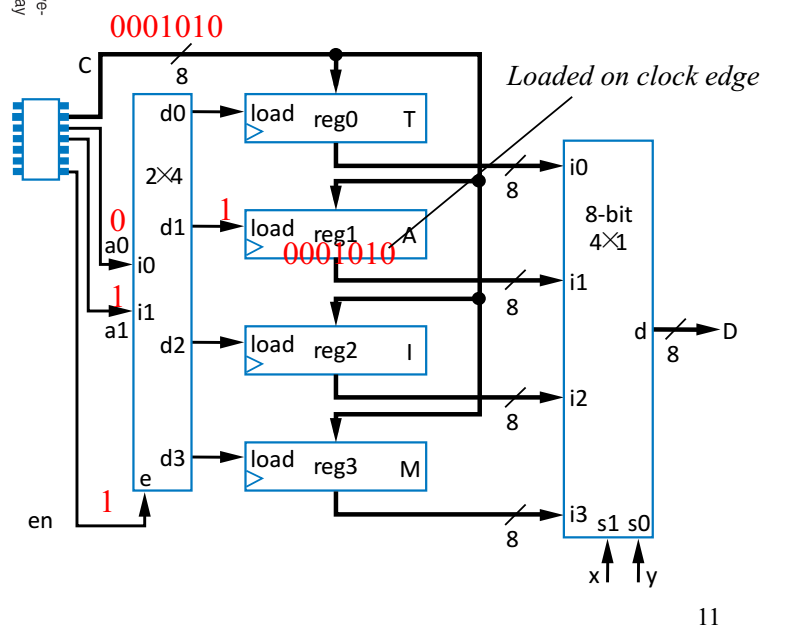


1 用了一个mux后可以选择是继续原来值还是载入新值，方便比较present value 和next value.

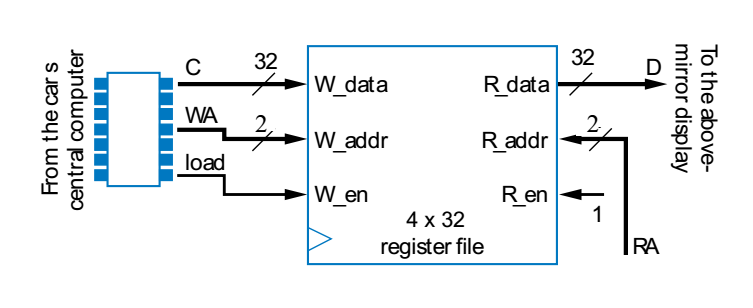


2 在车载显示屏的例子中：用一个mux来选择把值给哪一个load. 好处在于原本若是要显示不同的信息，例如四个信息需要分别对四个register供线，现在只需要一个公共线，mux选择数据要不要载入。

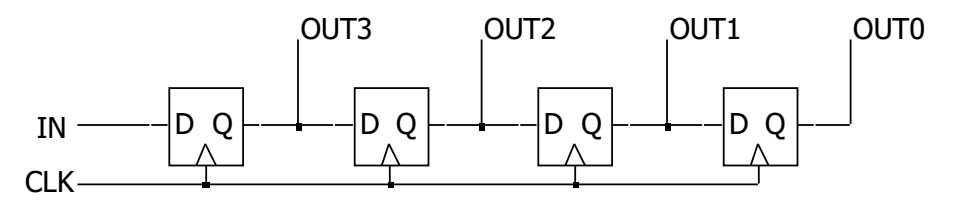
同时，在另一边还可以再加个mux选择哪个register里的信息能够被上传（read data）.

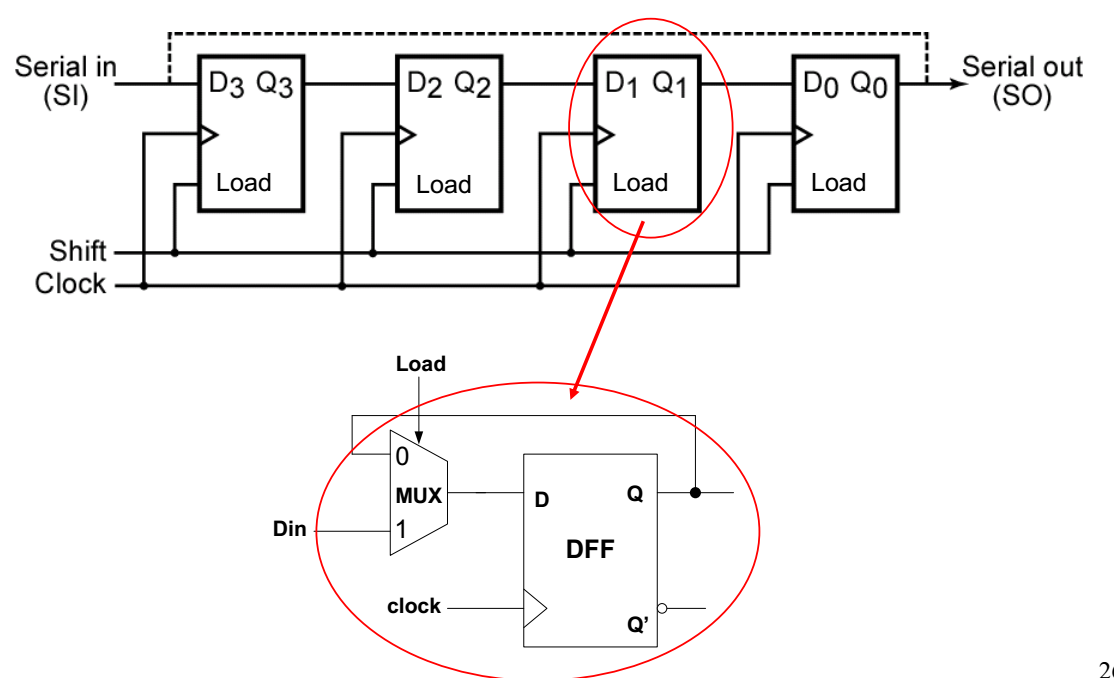


这种用两块mux和中间一堆register来存放不同的信息用来读写的被成为register file.

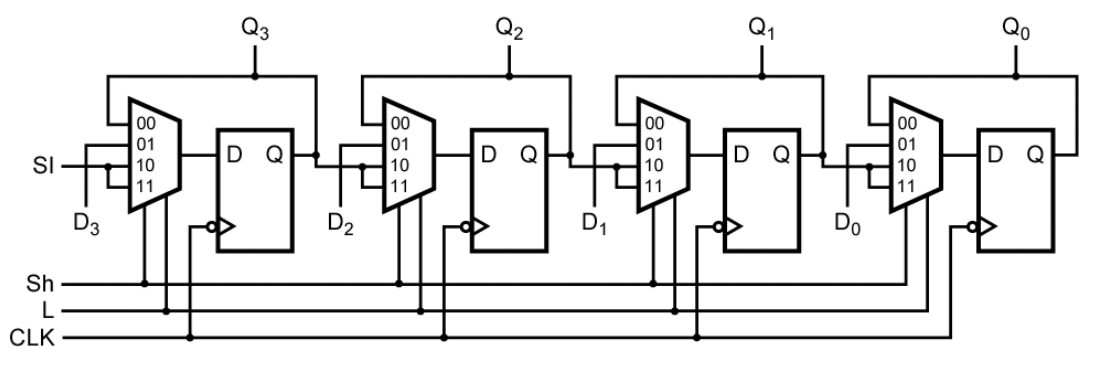


3 Shift register: 把信息bit by bit向一个方向移动



再给这些Dff加上一个load mux用来选择保持当前值还是继续

用更复杂一点的mux可以来要么shift要么load新值。

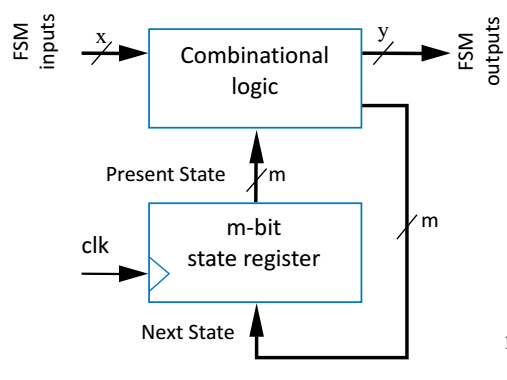


**Chapter 10 FSM**

性质：one condition must be true; All condition must be considered when leaving a state.

FSM的现实结构：（两部分）

1 State register; 2 Combinational logic. 合称是controller



设计FSM的五个步骤：

1 画FSM描述controller的behavior

2 做出结构：看需要多大的FF作为state register; FSM的输出是组合逻辑；带上输入的combinational logic 作为state register的bits和FSM的输入；combinational logic的输出作为next state bits 和FSM输出

3 给每个states编码

4 create the state table

5 根据以上实现combinational logic的部分

**Topic 11 FSM Optimization**

内容总览

优化方法：

1. State Reduction
2. Implication Tables
3. State encoding: One-hot encoding
4. Completing an FSM

FSM Reverse engineering.

1 State Reduction

做法：把state table列出来，要是next state, output都一样，那么把重复的那行删掉。接着把被删掉那行里输入无论是0/1的两种情况都改成与之重复的那个present state. 再由此用K-map化简。

2 Implication Tables

横向竖向分别写从state0-max

只保留下三角，对角线也删掉

看每个pair,如果output不一致，这一格打叉。

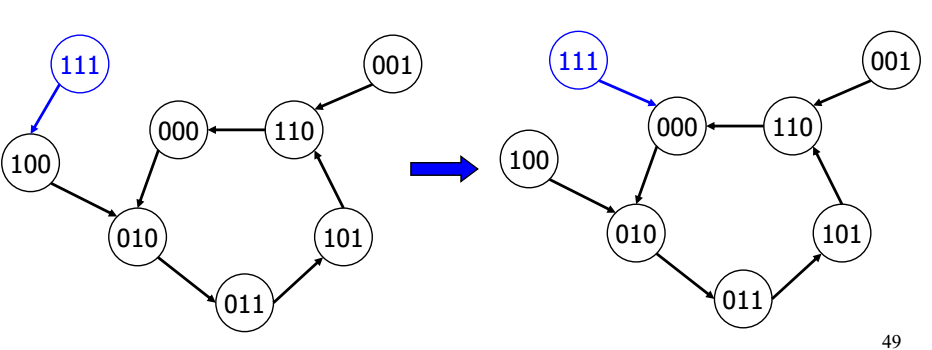
把叉全删掉后，剩下格子里写上当input=1时，两个的next state作为一个pair,总共有两个pairs.

要是这两个pair和别的剩下格子里的next state pair不相同，那打叉，要是相同（只是循序不一样），那么视为这两个是相同的。合并为一个

3 State Encoding: One-hot encoding

原本用00/01/10/11换成0001/0010/0100/1000

4 Complete FSM

把原本没有的状态X全写出来，用K-map化简，可能导致一些unwanted state返回wanted state 速度变慢，所以人为赋给这个state一个next state

FSM Reverse Engineering (Very important)

给一个FSM的电路，figure out the behavior

看以下几个方面：1 Mealy or Moore? / 2 How many states/ 3 State table? /4 State diagram?

Topic 12 RTL Design

RTL Design包含: Controller; Datapath components; Digital device.

RTL Design 方法：

1 Capture a high-level state machine

2 Create a Datapath

3 Connect the datapath to a controller

4 Derive the controller’s FSM

Topic 13

ALU; Carry lookahead adder; Incrementor; Multiplier

ALU: 可以执行各种各样的操作取决于当前的mux 选择哪一个；

实现方法：用先把数据进来，所有的操作全分别做完，由mux选择出哪一个作为output

Carry lookahead adder: