

基于FSM的交通灯控制

内容大纲

- **FSM概述**
- **FSM分类与异同**
- **FSM设计一般步骤与实例**
 - **序列检测器**
 - **交通灯**

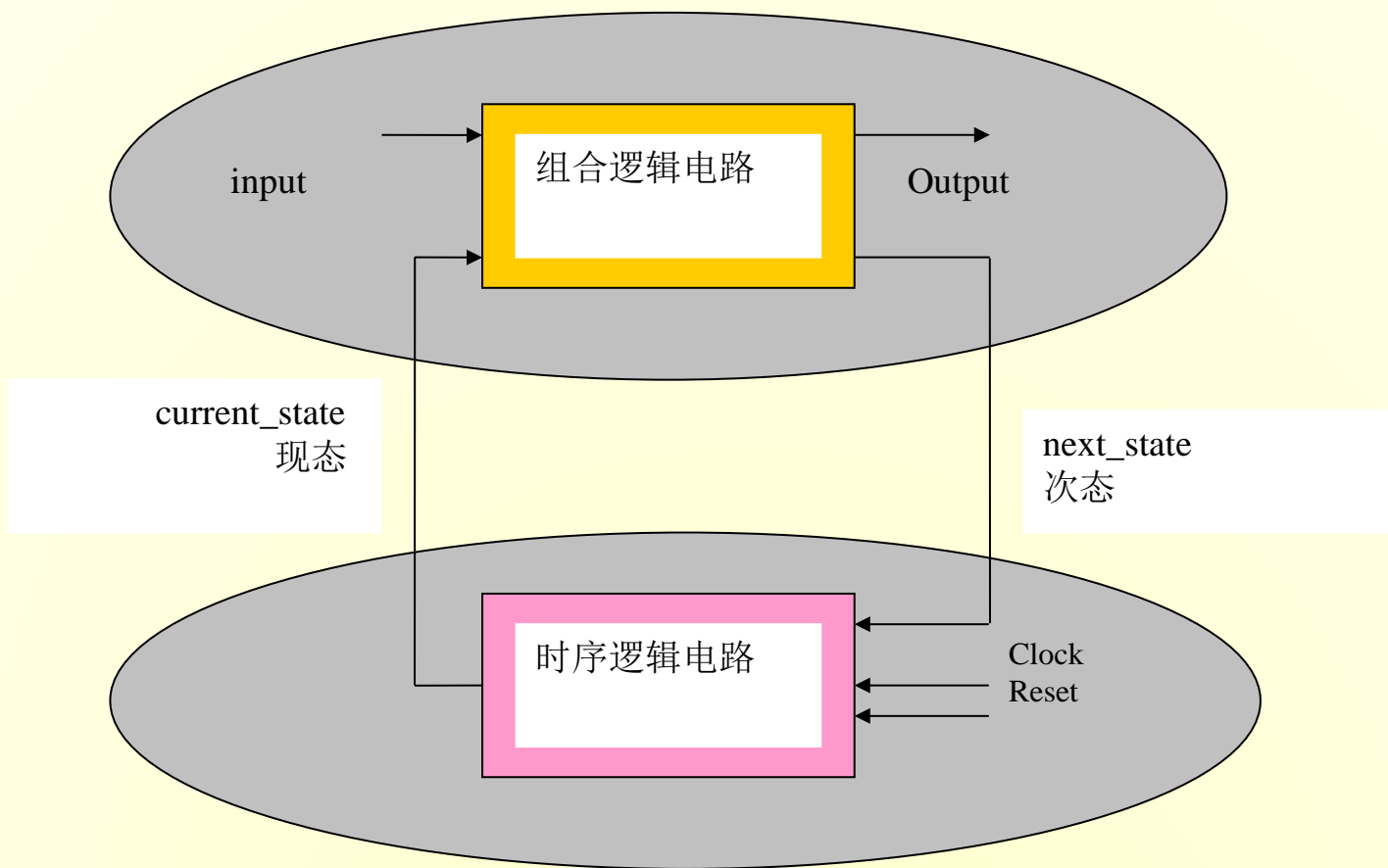
FSM概念

- ◆ FSM : finite state machine , 一种由寄存器组和组合逻辑构成重要的时序电路 , 在数字系统设计中有着非常重要的地位和作用。在同一时钟跳变沿由一个状态转移到另一个状态。包含时序、组合逻辑电路
- ◆ 非常有用模型 , 可以模拟大部分事物。如按键命令、自动门控制、通信时序等

1. 状态总数state是有限的
2. 任何一个时刻 , 只能处于一个状态
3. 在条件满足时 , 由一个状态转变到另一个状态

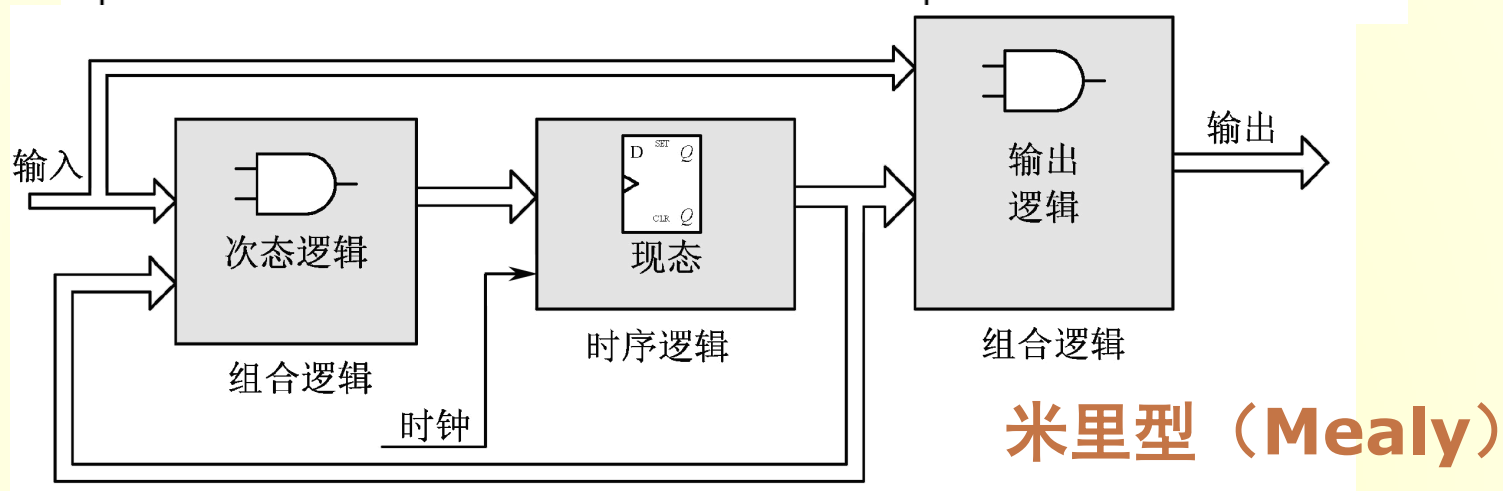
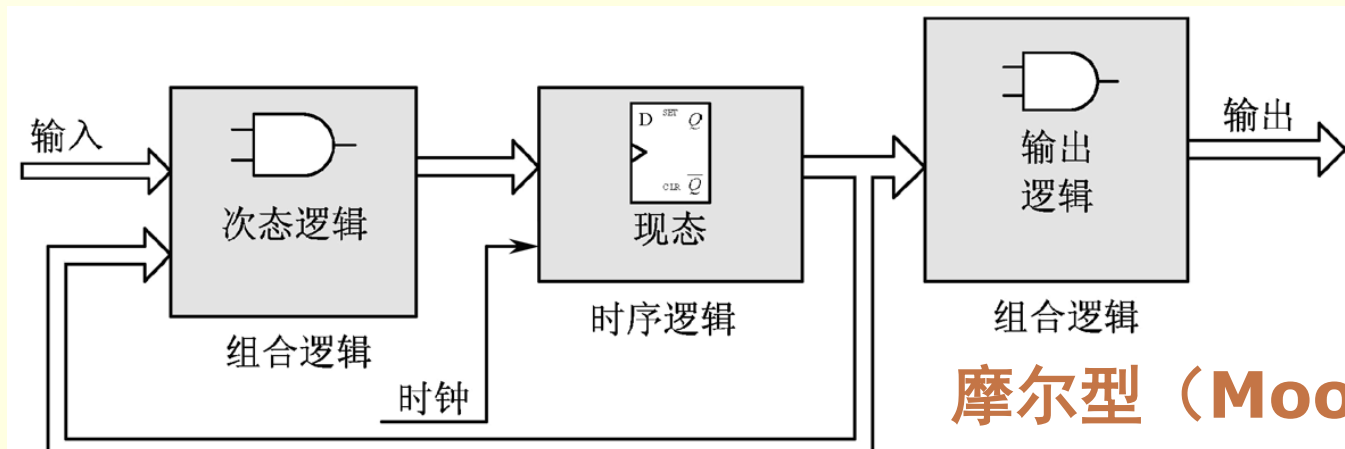
FSM概念

状态机电路结构图



FSM基本类型

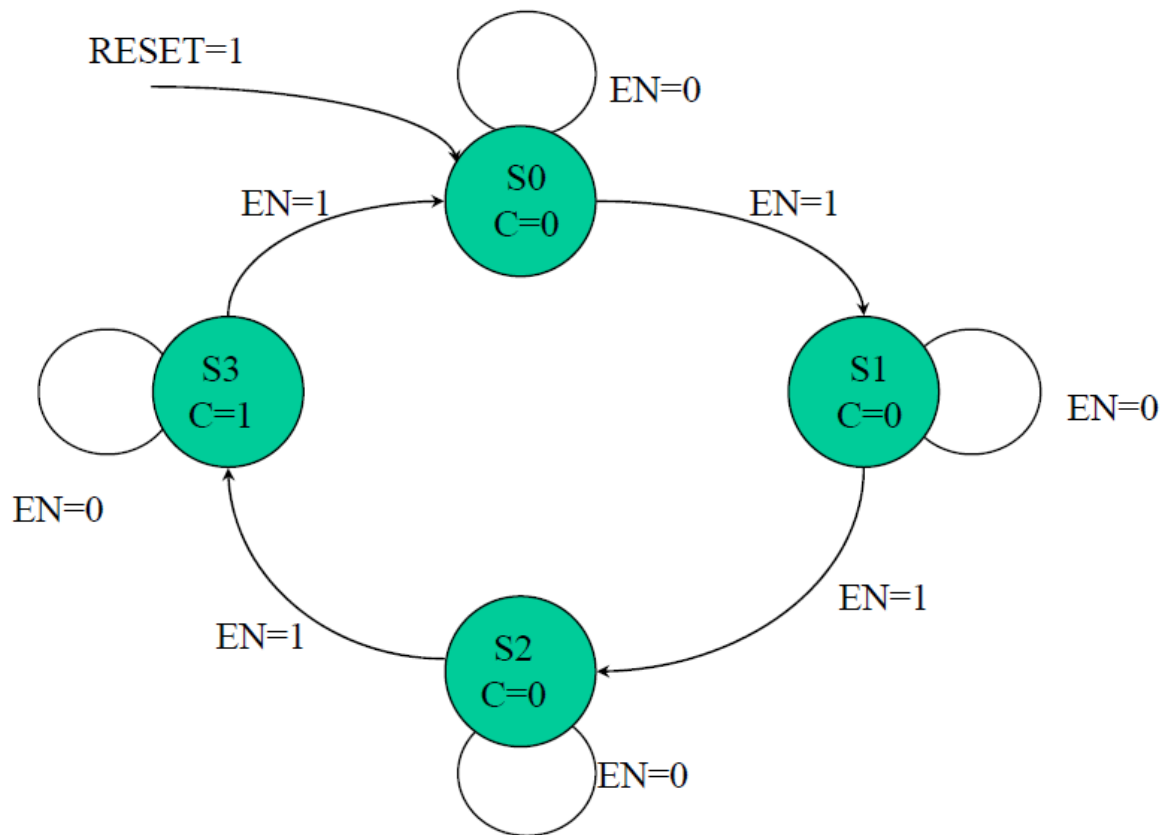
- ◆ Moore型：输出信号仅与现态相关
- ◆ Mealy型：输出信号与现态和输入相关



FSM基本类型

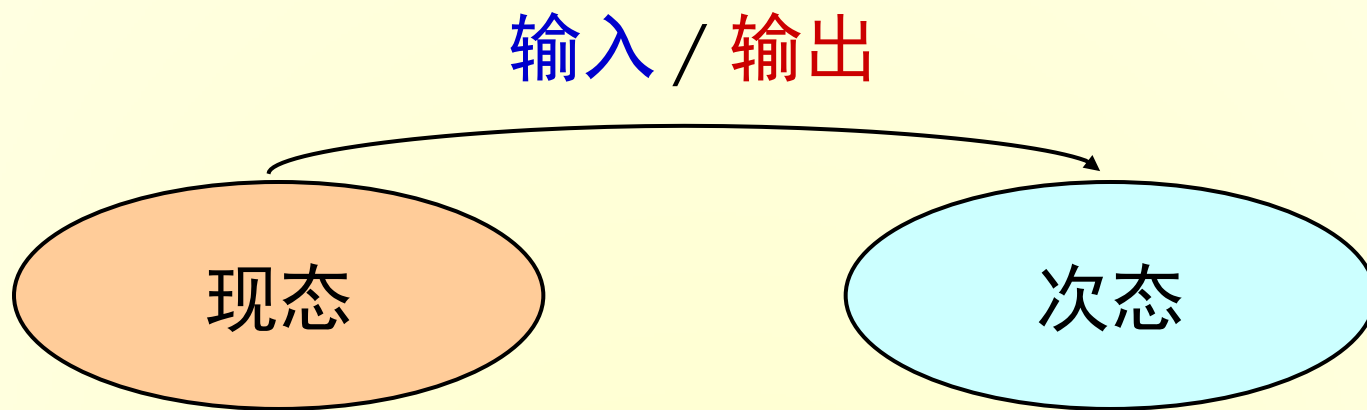
FSM通过状态图描述状态状态转移过程

Simple State Machine Example



FSM三要素

- 状态：当前态、次态
- 输入：触发状态转移的输入条件
- 输出：该状态所触发的输出动作



FSM编码

- ◆ 二进制码
- ◆ 格雷码
- ◆ 独热码

二进制编码、格雷码编码使用最少的触发器，消耗较多的组合逻辑，而独热码编码反之。

在CPLD中，由于器件拥有较多的地提供组合逻辑资源，所以CPLD多使用二进制编码或格雷码，而FPGA更多地提供触发器资源，所以在FPGA中多使用独热码编码，并非绝对，以实际应用而定。

二进制	格雷码	独热码
000	000	00000001
001	001	00000010
010	011	00000100
011	010	00001000
100	110	00010000
101	111	00100000
110	101	01000000
111	100	10000000

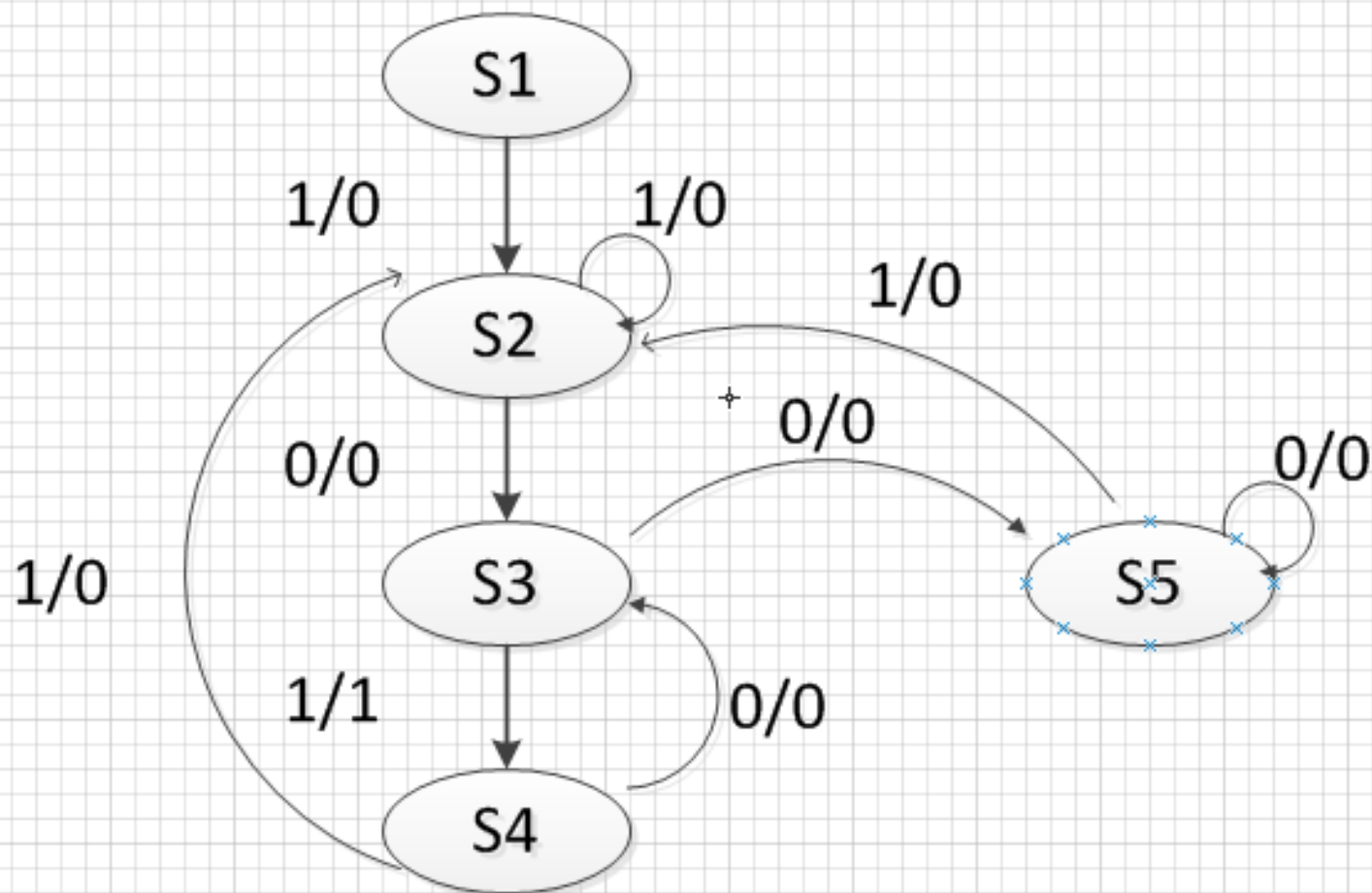
举例：序列检测器

序列检测器是将一个指定二进制序列从数码流中检测出来

如：将“101”序列从码流“111010110”中检测，输出高代表检测到序列，低电平代表没有发现

CLK	1	2	3	4	5	6	7	8	9
IN	1	1	1	0	1	0	1	1	0
OUT	0	0	0	0	1	0	1	0	0

序列检测器状态图



FSM设计的一般步骤

1. 逻辑抽象，得出状态转换图
2. 状态化简，如果在状态转换图中出现这样两个状态，它们在相同的输入下转换到同一状态去，并得到一样的输出，则称它们为等价状态，可合并成一个。
3. 状态分配：又称状态编码
4. 描述状态机
(选定触发器的类型并求出状态方程、驱动方程和输出方程
按照方程得出逻辑图)

FSM的描述方法

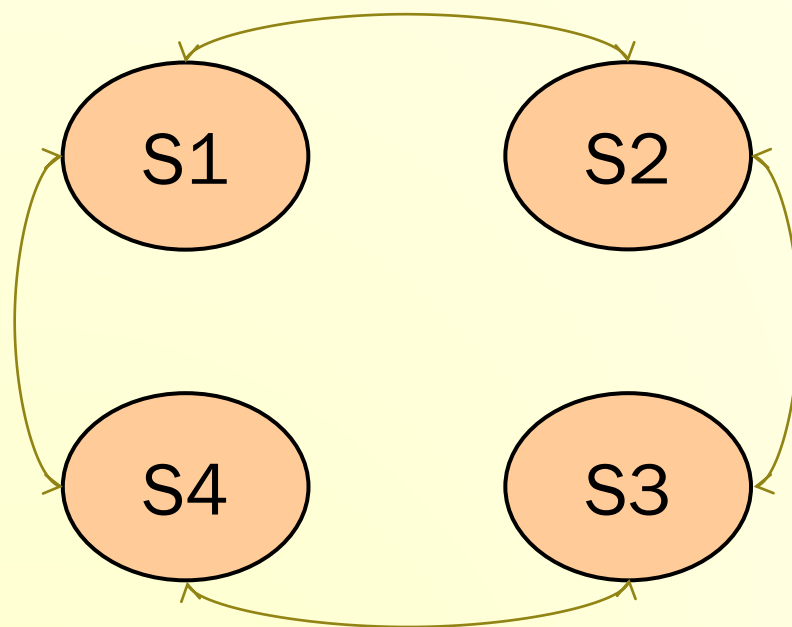
1. **一段式**——整个状态机在一个进程中，既包含状态转移也含有组合逻辑输入输出。简单，代码难维护。
2. **二段式**——两个进程，一个实现状态转移，一个实现组合逻辑。需要定义两个状态——现态和次态
3. **三段式**——三个进程，一个实现状态转移，一个实现组合逻辑，一个实现状态同步输出

举例：简易交通灯

十字路口简易交通灯，更具两个方向交通灯颜色变化能组成四种状态。根据这四种状态我们能够进行二进制编码。

状态转移的条件是根据十字路口红绿灯的计时

	横向	纵向	编码
S1	绿	红	00
S2	黄	红	01
S3	红	绿	10
S4	红	黄	11



交通灯的一段式描述

定义编码和状态变量

状态转移，输入条件判断，输出的动作在一个进程（ process ）里进行。组合逻辑和时序逻辑混合输出

交通灯的二段式描述

定义编码和状态变量，状态变量分为两个当前态和次态

程序包括两个进程模块，状态的转移用时钟同步，时序逻辑。

输入条件判断和输出动作在同一个组合逻辑里输出

交通灯的三段式描述

定义编码和状态变量，状态变量分为两个当前态和次态。和二段式一样

一个进程同步状态的转移，时序逻辑

第二个进程根据输入条件判断次态，组合逻辑

第三个进程同步次态下输出的动作，时序逻辑

FSM设计的要点

- ◆ 状态的提取要完全
- ◆ 状态编码正确，保证状态机能正常跳转
- ◆ 完整状态机需要初始态和默认状态
- ◆ 常在进程里用case
- ◆ Case必须包含默认态default
- ◆ 进程带有边沿信号列表表示时序逻辑，否则是组合逻辑
- ◆ 每个进程只能由一个时钟沿跳变
- ◆ 同一信号不能在不同进程下赋值