EECS 151/251 A Discussion 5

Feb 16, 2024

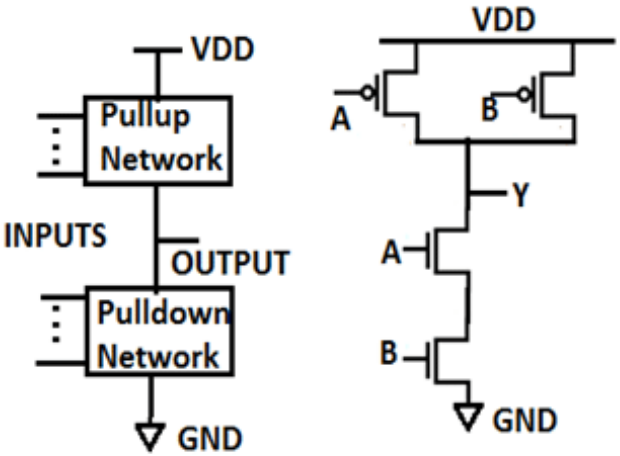
Berkeley
UNIVERSITY OF CALIFORNIA

Content

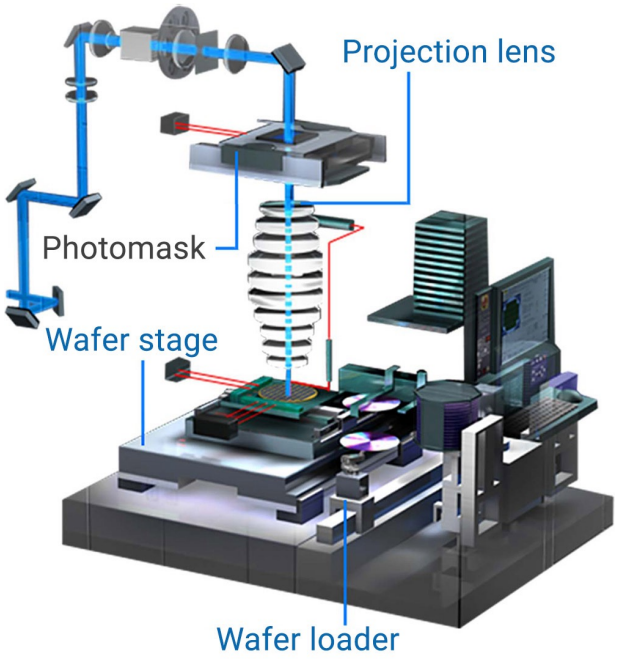
- CMOS circuits
- Fabrication
- FSMs
- Problems

CMOS Circuits

- **Complementary metal-oxide-semiconductor (CMOS) is a semiconductor manufacturing process which creates symmetric n-type and p-type transistor**
  - PMOS are weaker, than NMOS
  - Pull-up and Pull-down network
- Pull-up network is circuit representing Boolean function is logic high; created with PMOS
- Pull-down network is circuit representing Boolean functions is logic low; created with NMOS
- Pull-up and pull-down are complement circuits related through DeMorgan's Law



Fabrication



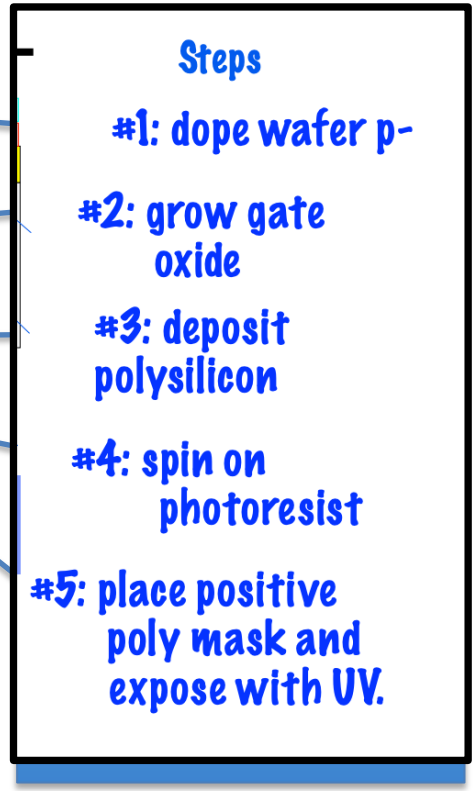
http://tinyurl.com/2bvv6pta

- Fabrication is the manufacturing of ICs
- *Foundry ("Fab") – "A semiconductor manufacturer that makes chips for other companies"*
  - Provides PDK
  - Popular Fabs: TSMC, Global Foundries, Samsung, Intel (soon…Chips Act)
- Different methods to create CMOS: single well (n-well or p-well, dual well (twin tub), triple tub, etc
  - **Single n-well is what is presented in lecture slides**
- Terminology:
  - *Oxide* – SiO_2 layer formed exposing pure silicon wafer to oxygen
  - *Polysilicon* – crystalline silicon material used to form gate
  - *Metallization* – Add metal layers and transistor contacts
  - *Photomasks ("masks" or reticle)* – an opaque glass substrate with a single layer of IC etched onto it so light passes through in etched areas
  - *Lithography* – "*a photographic process by which a light-sensitive polymer, called a photoresist, is exposed and developed to form 3D relief images on the substrate.*"
    - State-of-art use is EUV (13.5nm) for fine detail (5nm and 3nm)
    - ASML is the #1 manufacture of lithography machines; Only company with EUV machines on market

Fabrication: Steps

1. Grow Silicon ignot
2. Slice ignot into wafers (at this point it's just the substrate)
3. Dope the entire wafer (substrate) with p- or n+
4. Grow oxide on surface, etch away part of oxide for space to create wells
5. Create wells (either n-well or p-wells)
6. Grow oxide again (for gate)
7. Place film of polysilicon on surface
8. Add photoresist where gates should be and expose to UV
9. Chemical etch away unwanted poly and oxide
10. Add for source and drains for both NMOS and PMOS with correct dopant
11. Grow another (thick) oxide layer and add contacts (metallization)
12. More masking and remove extra metal (repeat for all metal layers)

Simplified steps from Lecture:



Fabrication: Key Facts

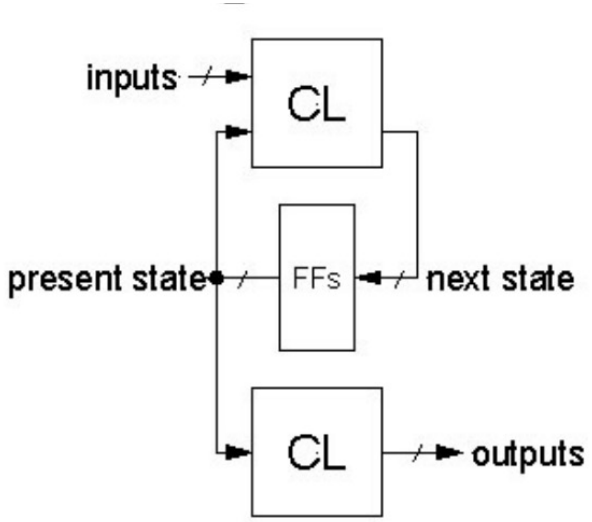
1. CMOS is modern process for ICs
2. The steps for single well fabrication (from lecture)
3. ICs and any ASIC is not 2D, but a 3D layer cake
4. ASML makes EUV lithography machines which makes small nodes possible
5. Which dopant for NMOS? Arsenic
6. Which dopant for PMOS? (find yourself)
7. A reticle, or mask, is used to etch the design layer by layer
8. Fabs receive design as (usually) a GSD file from design house and manufacture the semiconductor
9. Currently smallest nodes: 5nm and 3nm

Finite State Machine (Part 2)

- FSMs orchestrate a sequence of events throughout *time*
  - Must use registers
- State diagrams visualize describe FSMs
  - States are typically gray coded
  - Arrows between state represents transition
    - Transitions without conditions are taken regardless
    - Transitions with conditions are only taken if condition is met
- Two defined types (difference is when the output changes)
  - Moore: the output is dependent solely on the state
  - Mealy: the output is dependent the input and the state
  - **In practice, many FSMs are a mixture of both**

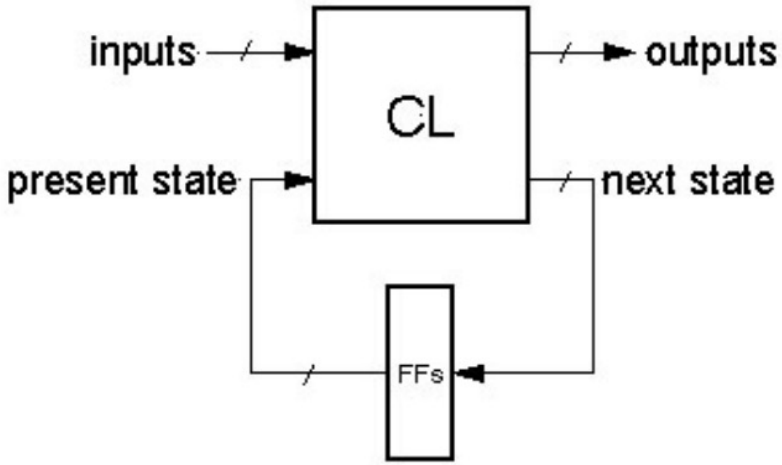
Finite State Machine: Moore

- Definition: the output is dependent solely on the state
  - In laymen's terms, the output can not change on state transition



Finite State Machine: Mealy

- Definition: the output is dependent the input and the state
  - In laymen's terms, Output can change of the transition



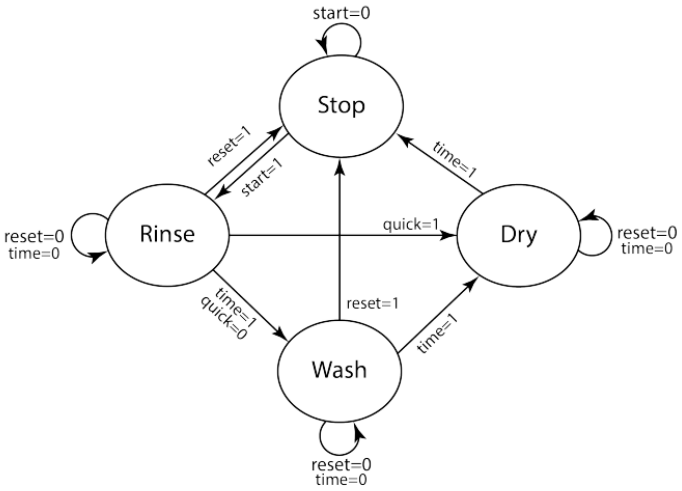
Finite State Machine: Verilog

- FSMs orchestrates a sequence of events throughout *time* by maintaining state (must use registers)
- Two styles for FSM in HDL
  1. Two process (one sequential, one combinational)
  2. One process containing both sequential and combinational logic

Problems

Problem 1. FSM Design (HW4 Spring 4)

Consider a simple dishwasher that has an option for a quick rinse cycle or a full wash cycle. The dishwasher will first rinse the dishes with hot water, then decide whether to wash with detergent or skip straight to drying depending on the setting. If you open the door of the dishwasher prematurely, a door switch will trigger a `reset` and the dishwasher will stop and go back to the initial state. A timer raises a `time` flag when the dishwasher is ready to move to the next state. The dishwasher has three outputs: `water`, `detergent`, `heat`. During the rinse cycle, it will request water. During the wash cycle, it will request water and detergent. During the dry cycle it will request only heat. Here is a conceptual state transition diagram for the dishwasher:



For each of the following scenarios, please provide:

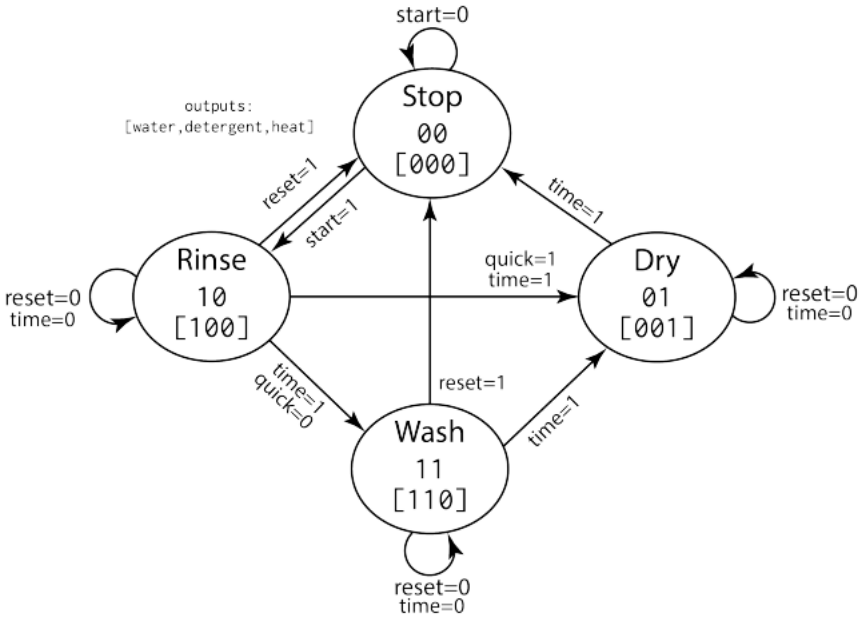
- A state transition diagram with the state names and encoding (e.g. `STOP (00)`), as well as outputs labeled appropriately.
- A circuit diagram of the state machine. The state machine will receive the inputs `reset`, `start`, `quick`, and `time`. The state machine must have the outputs `water`, `detergent`, and `heat`. You may use logic gates of 4 inputs or fewer as well as multiplexers to implement your next state logic.
- A quick (1-2 sentence) summary of your design process and decisions.

Design the FSM for each of these scenarios:

(a) Binary-encoded Moore Machine

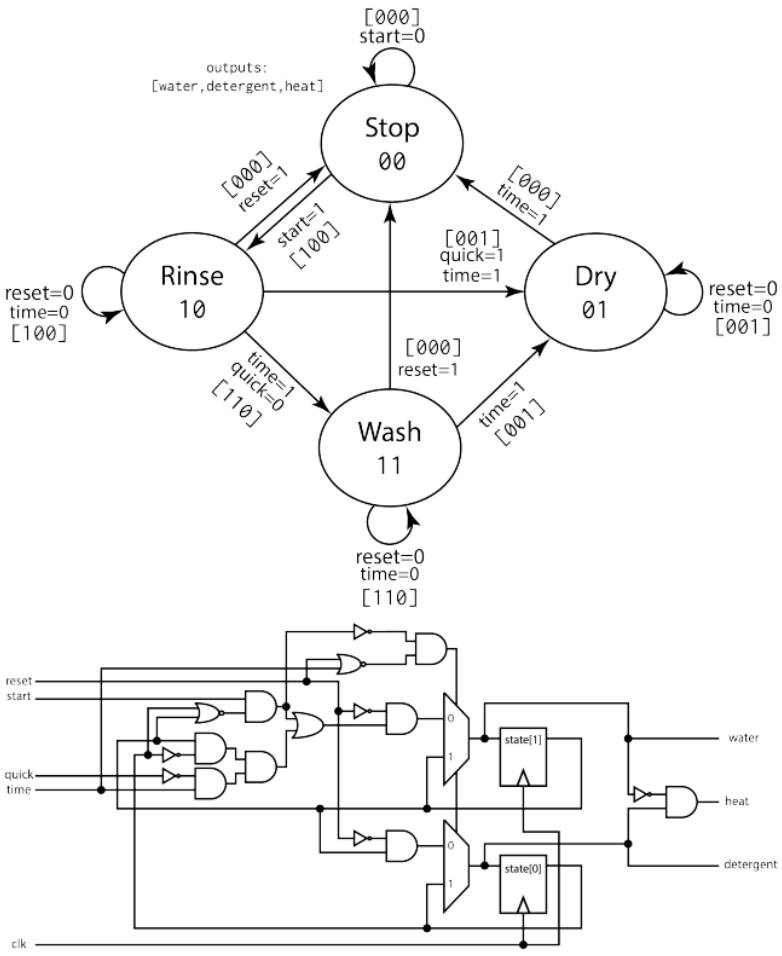
Solution:

There are many approaches to this problem. In this case, I have purposely encoded my states so that the output logic is relatively simple.



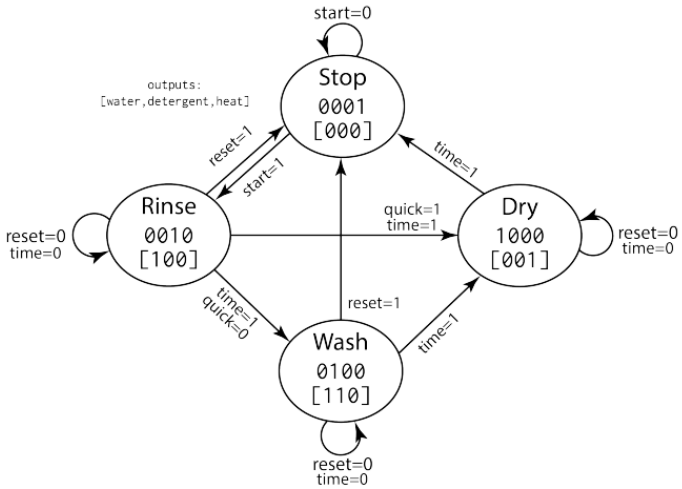
(b) Binary-encoded Mealy Machine

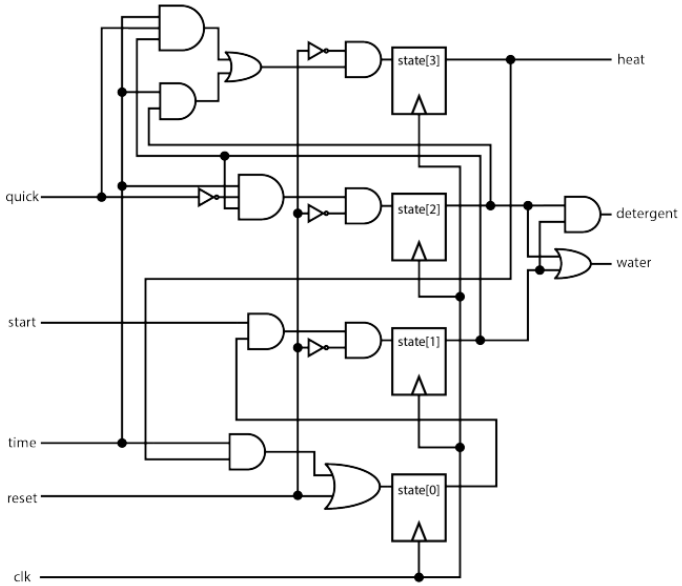
Solution:



(c) One-hot Moore Machine

Solution:





(d) One-hot Mealy Machine

Solution: