Administrative issues

- Midterm #1 will be given Tuesday, October 29, at 9:30am.
  - The entire class period (75 minutes) will be used.
  - Open book, open notes.
  - *DDPP* sections: 2.1–2.6, 2.10–2.13, 3.1–3.4, 3.7, 4.1–4.3.7, 4.5, 5.1–5.2, 5.4–5.5, 5.7, 5.9, 6.1.
- Lab #5 will be handed out next week and due November 6–7.

  No laboratory assignment next week. :-(

Sequential logic

Executive summary of EE 121: combinational logic easy, sequential logic hard.

Combinational logic = truth tables.

Combinational logic synthesis can be greatly aided by clever decomposition of the problem. Example: 8-bit adder built using two 4-bit adders.

Sequential logic = simple matter of programming. ;-)

Designing state machines is similar to programming, but much harder because things happen in parallel.

In software programs, data is stored in *variables*—simple state machines that remember one or more bits. A stored value remains unchanged until a new value is loaded. The values of the variables are *part* of the state of the system.

State machine examples

"Complex" state machines:

- Record/play control for digital audio (EE 121 lab #6)
- Control unit for computer (very ambitious final project)

Examples of small state machines:

- Light switch controller (toggle on/off)
- Simultaneous button push detector
- Push button processor converts long button push to one clock cycle pulse

What these examples have in common: the output is a function of past inputs (memory) as well as current input.

State machines

A state machine is a sequential circuit whose outputs depend on the current state (values of memory devices) and whose state changes based on current state and inputs.

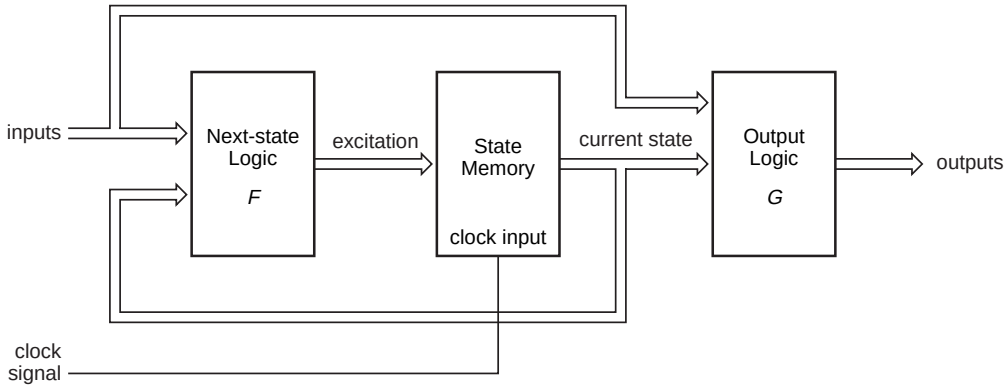
state $=$ F(state, input) — next state function
output $=$ G(state, input) — output function (may not depend on input)

State machines may be asynchronous or synchonous:

- State transitions may occur at any time when input changes
  Examples: R-S latch, D latch
- State transitions occur only at times determined by system clock, usually at active an clock edge.
  Examples: D flip-flop, shift register = series of D flip-flops

Clocked synchronous state machines: Mealy

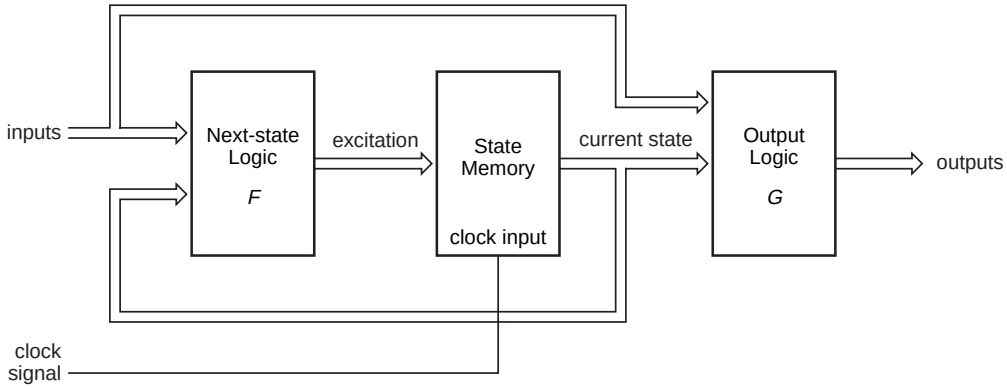
Mealy machine: output depends on state and current input.



George Mealy worked at Bell Labs in the 1950s.

Clocked synchronous state machines: Moore

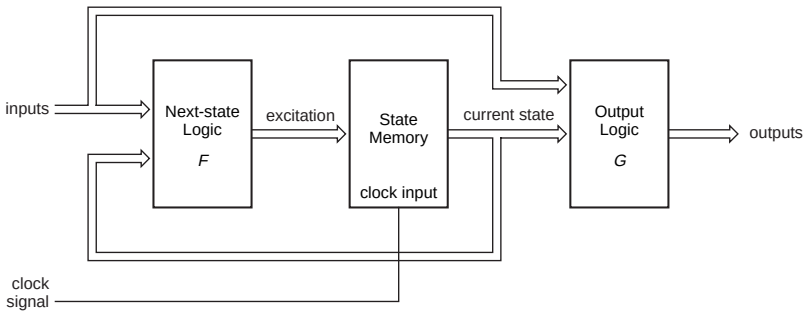
Moore machine: output depends only on current state.



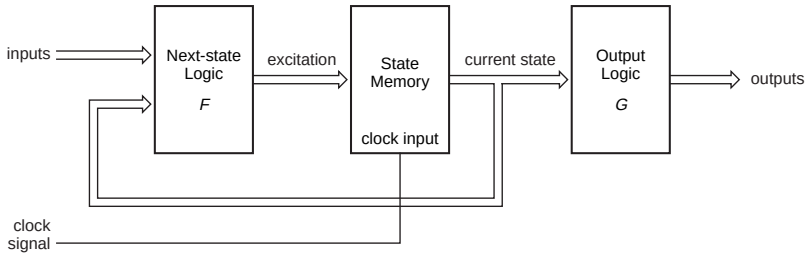
E.F. Moore was Electrical Engineering professor at the University of Pennsylvania after working at Bell Labs.

Mealy vs. Moore

Mealy:



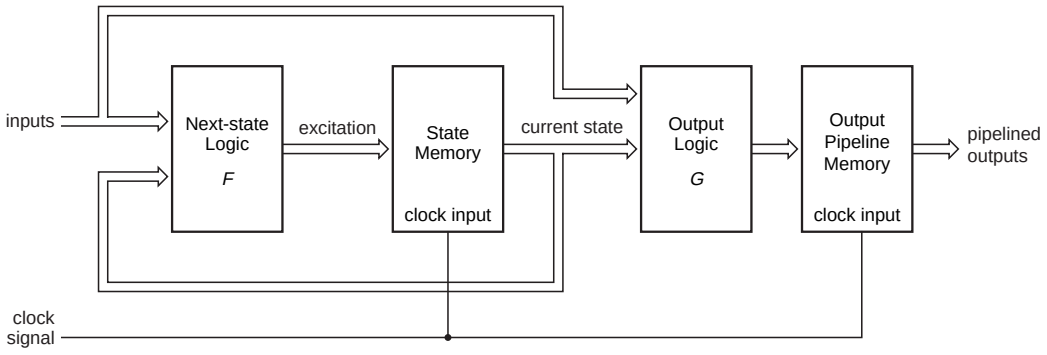
Moore:



Mealy machines are more powerful, but Moore machines are easier.

Mealy machine with pipelined outputs

Outputs of a Mealy machine can be kept constant within a clock period by using output flip-flops.



Drawback: output changes are delayed by as much as one clock cycle.

Characteristic equations

State machine memory is built using latches or flip-flops.

Each device has a *characteristic equation* that describes how the state machine changes state as a function of the inputs.

Device Type	*Characteristic Equation*
S-R latch	$Q* = S + R' \cdot Q$
D latch	$Q* = D$
Edge-triggered D flip-flop	$Q* = D$
D flip-flop with enable	$Q* = EN \cdot D + EN' \cdot Q$
Master/slave S-R flip-flop	$Q* = S + R' \cdot Q$
Master/slave J-K flip-flop	$Q* = J \cdot Q' + K' \cdot Q$
Edge-triggered J-K flip-flop	$Q* = J \cdot Q' + K' \cdot Q$
T flip-flop	$Q* = Q'$
T flip-flop with enable	$Q* = EN \cdot Q' + EN' \cdot Q$

Table 7-1 Latch and flip-flop characteristic equations.

The characteristic equation of the D flip-flop is very simple, so we use it!

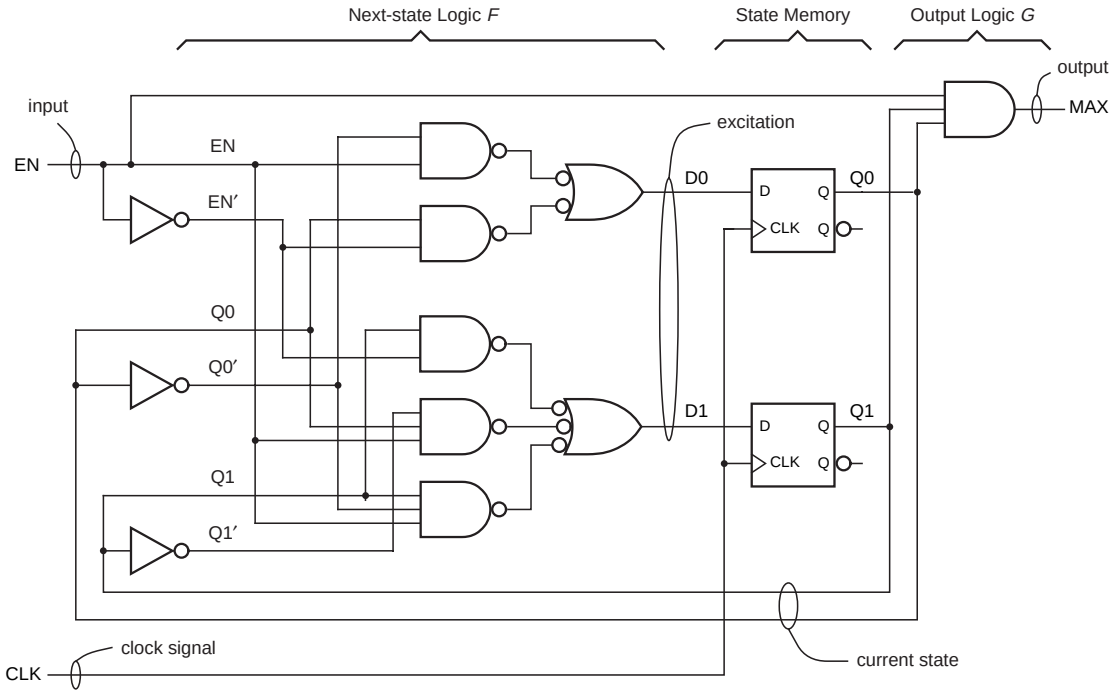
Clocked synchronous state machine analysis

Clocked synchronous state machines can be described in many ways:

- circuit schematic
- state and state/output tables
- transition and transition/output tables
- state diagrams (flowcharts)
- ASM (algorithmic state machine) charts
- HDL (hardware description languages)
- programming languages

A description that can be given to a CAD system for simulation and synthesis is preferred. Usually these are text descriptions, but drawing tools exist.

Clocked synchronous state machine example



State machine analysis

Excitation equations:

$$D0 = Q0 \cdot EN' + Q0' \cdot EN$$
$$D1 = Q1 \cdot EN' + Q1' \cdot Q0 \cdot EN + Q1 \cdot Q0' \cdot EN$$

Characteristic equations:

$$Q0^* = D0$$
$$Q1^* = D1$$

Transition equations:

$$Q0^* = Q0 \cdot EN' + Q0' \cdot EN$$
$$Q1^* = Q1 \cdot EN' + Q1' \cdot Q0 \cdot EN + Q1 \cdot Q0' \cdot EN$$

Transition, state, and state/output tables

(a)

	EN	
Q1 Q0	0	1
00	00	01
01	01	10
10	10	11
11	11	00
	Q1* Q0*	

(b)

	EN	
S	0	1
A	A	B
B	B	C
C	C	D
D	D	A
	S*	

(c)

	EN	
S	0	1
A	A, 0	B, 0
B	B, 0	C, 0
C	C, 0	D, 0
D	D, 0	A, 1
	S*, MAX	

Table 7-2 Transition, state state/output tabl the state machir Figure 7-38.

Output equation:

$$MAX = Q1 \cdot Q0 \cdot EN$$

State diagram and timing diagram