Sequential Circuits (2)

Prof. Usagi

Recap: Combinational v.s. sequential logic

- Combinational logic
  - The output is a pure function of its current inputs
  - The output doesn’t change regardless how many times the logic is triggered — Idempotent
- Sequential logic
  - The output depends on current inputs, previous inputs, their history

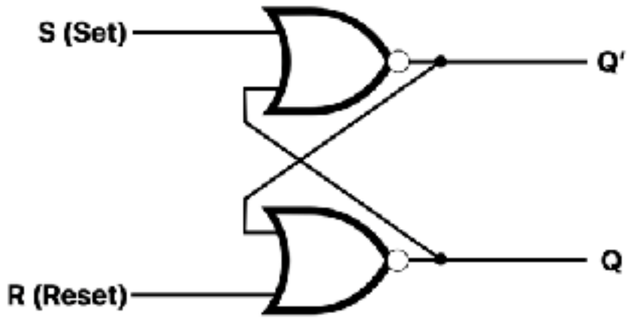
Sequential circuit has memory!

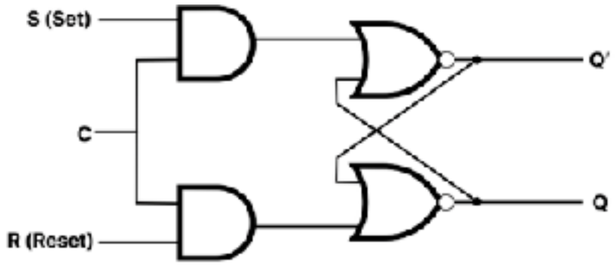
Recap: Theory behind each

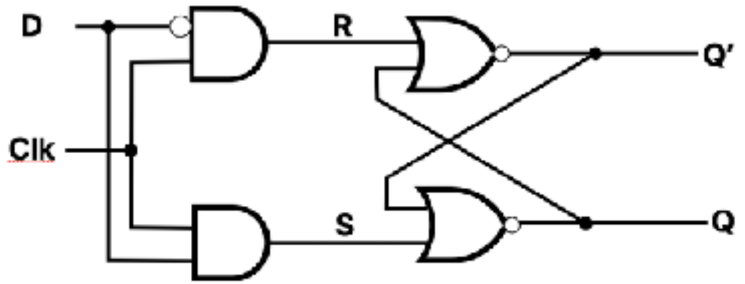
- A **Combinational logic** is the implementation of a **Boolean Algebra** function with only Boolean Variables as their inputs
- A **Sequential logic** is the implementation of a **Finite-State Machine**

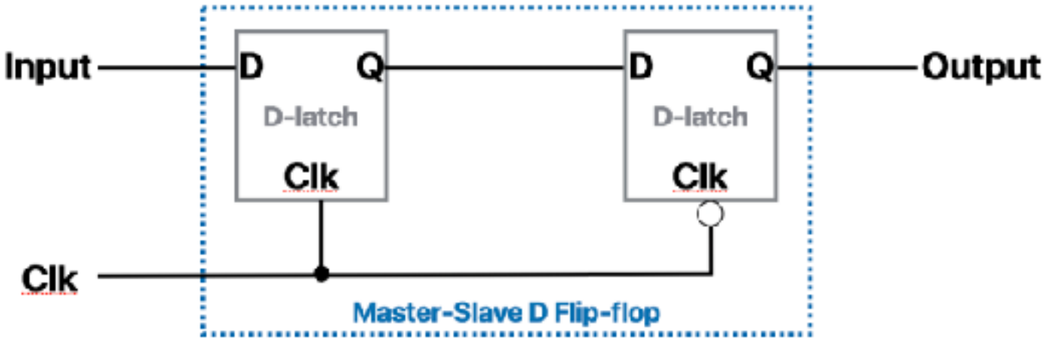
Recap: 4-different types of bit storage

- SR-latch
  - S = 1 sets Q = 1
  - R = 1 sets Q = 0
  - Problem: S = 1, R = 1, Q = undefined
- Level-sensitive SR-latch
  - S, R only become effective when C = 1
  - Problem: avoid the case of signal oscillation, but cannot avoid the "intensional" 1,1 inputs
- D-latch
  - SR can never be 11 if the Clk is set appropriately
  - Problem: D single needs to be stably long enough to set the memory
- D-flip-flop
  - Only loads the value into memory in the beginning of the rising edge. Values can hold for a complete clock cycle
  - Problem: more gates

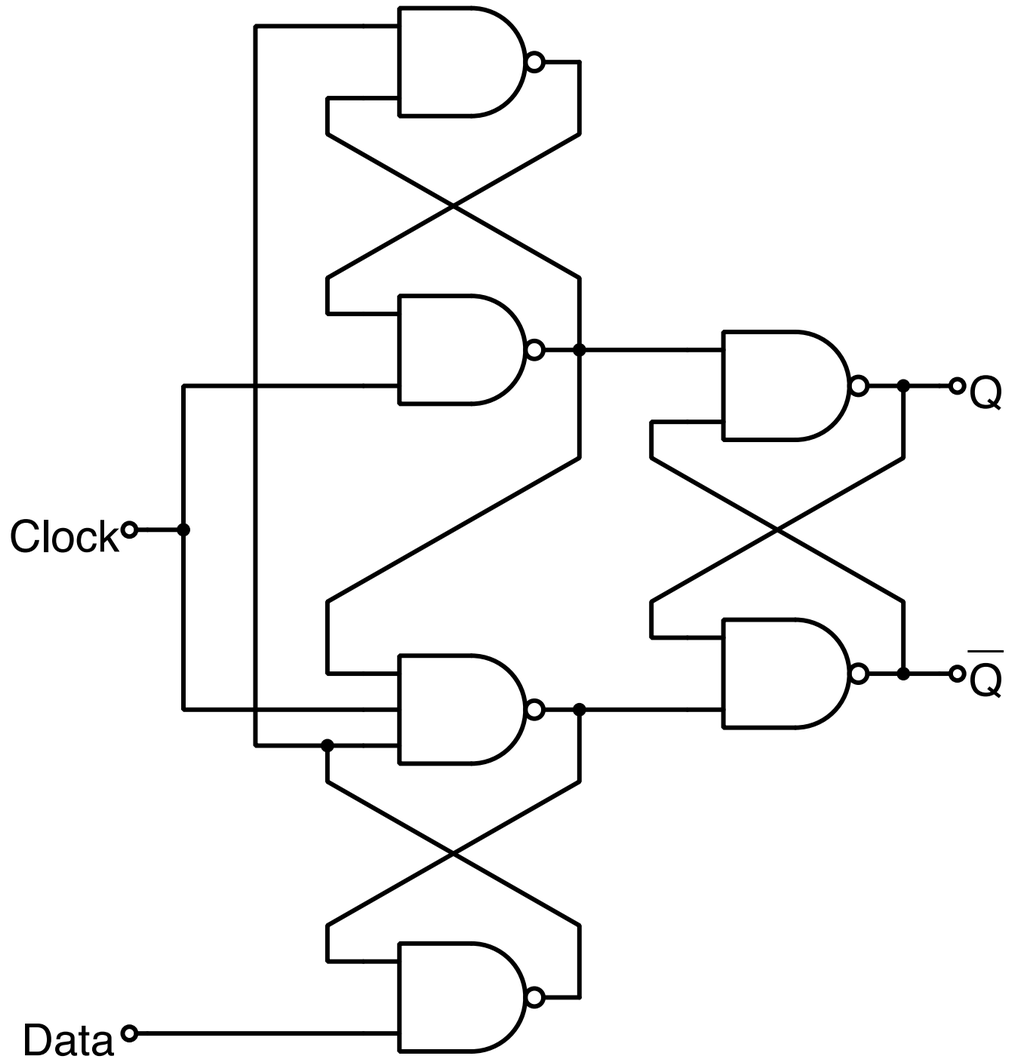




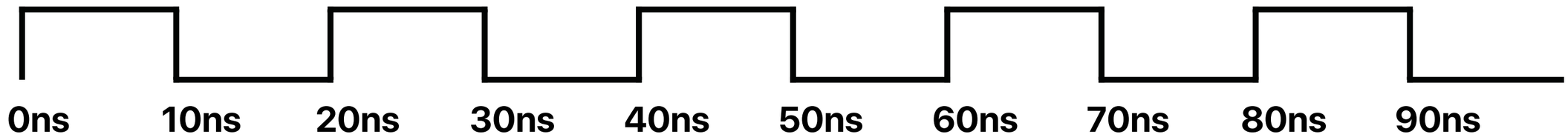




Positive-edge-triggered D flip-flop



Recap: Clock signal



- Clock -- Pulsing signal for enabling latches; ticks like a clock
- Synchronous circuit: sequential circuit with a clock
- Clock period: time between pulse starts
  - Above signal: period = 20 ns
- Clock cycle: one such time interval
  - Above signal shows 3.5 clock cycles
- Clock duty cycle: time clock is high
  - 50% in this case
- Clock frequency: 1/period
  - Above : freq = 1 / 20ns = 50MHz;

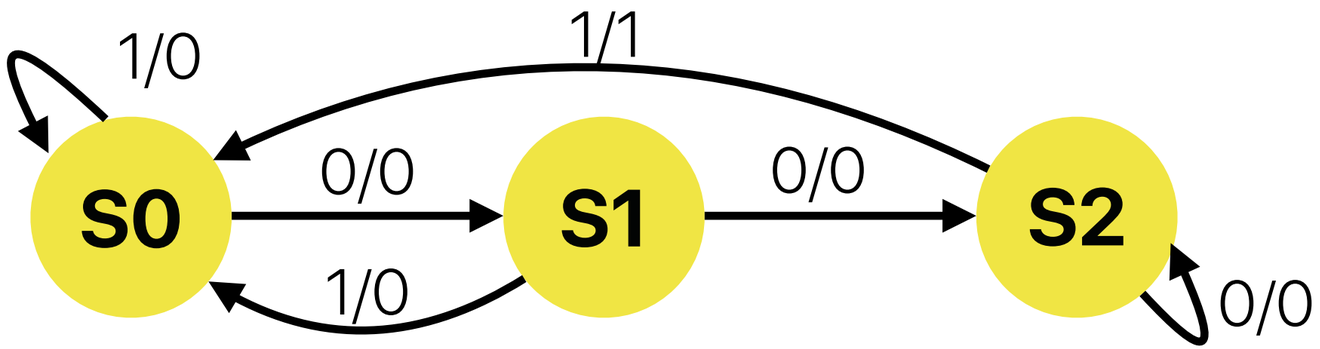
Outline

- From FSM to circuit
- Canonical forms of FSMs
- When sequential circuits meets datapath components

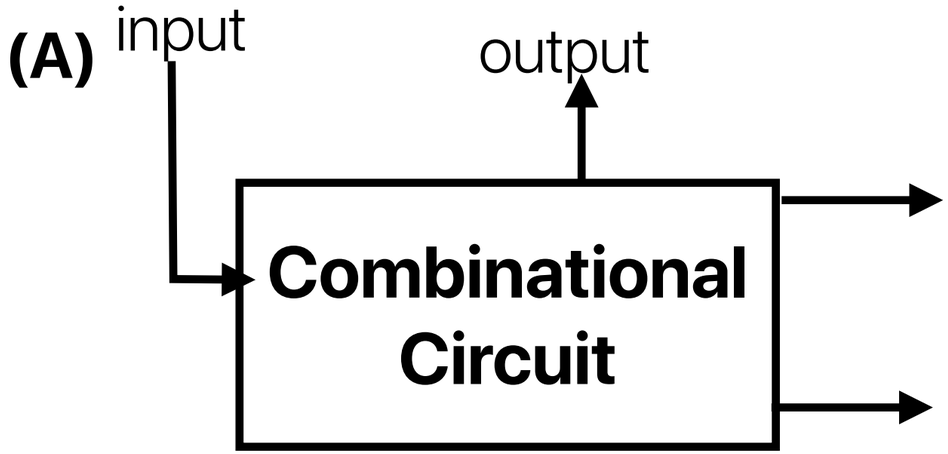
Let’s learn how to design sequential circuits!

Poll close in 1:30

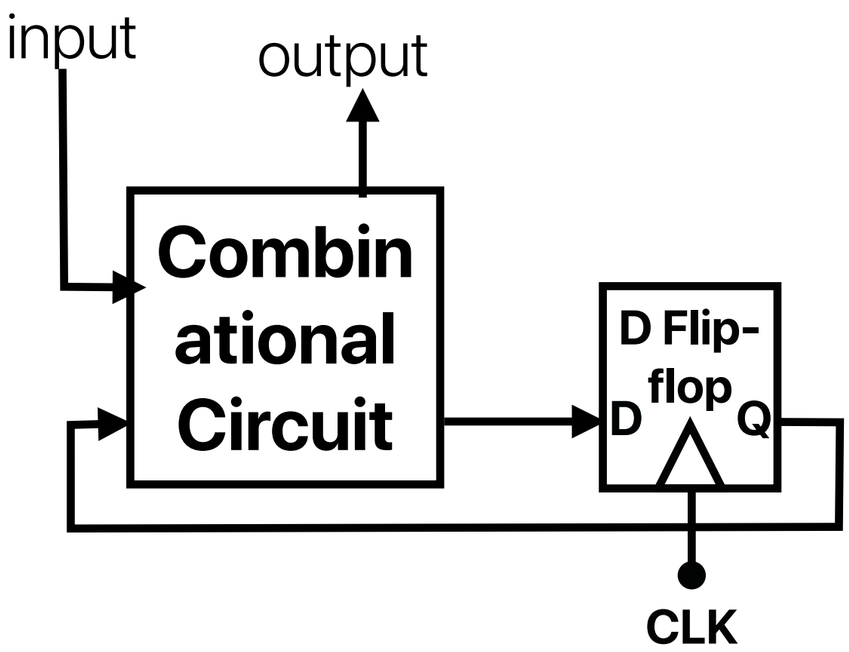
Which is the most likely circuit realization of Life on Mars



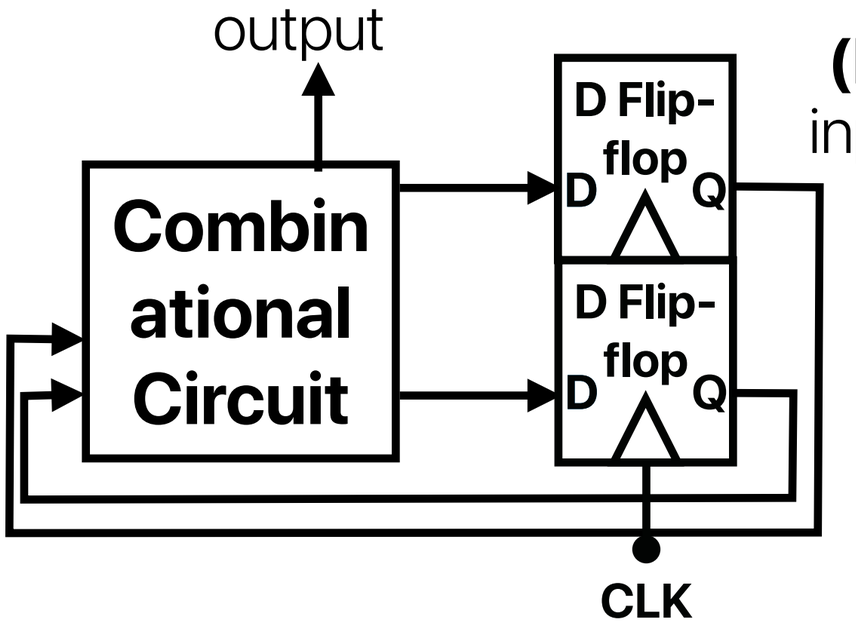
(A)



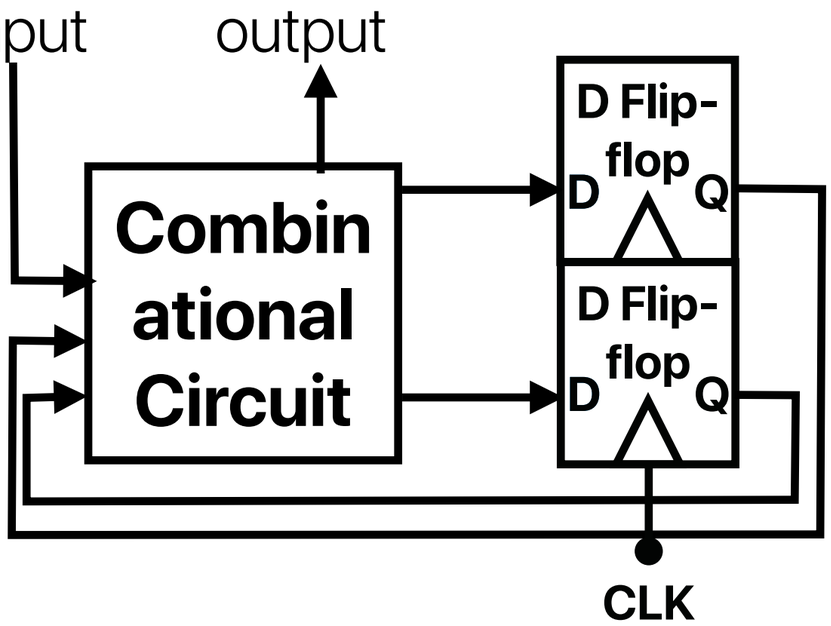
(B)



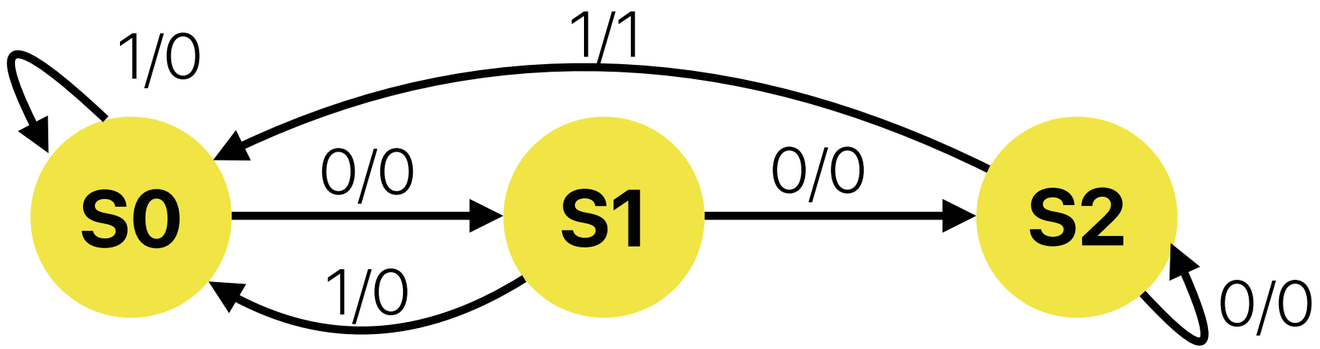
(C)



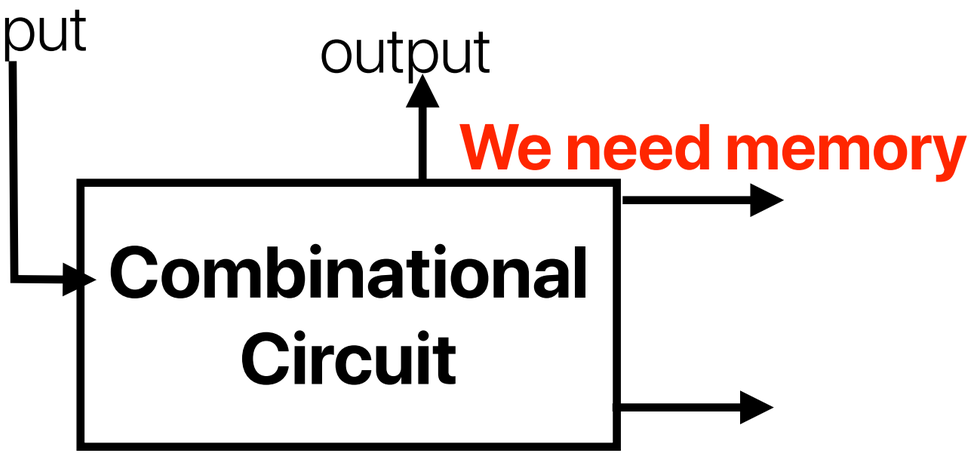
(D)



Which is the most likely circuit realization of Life on Mars

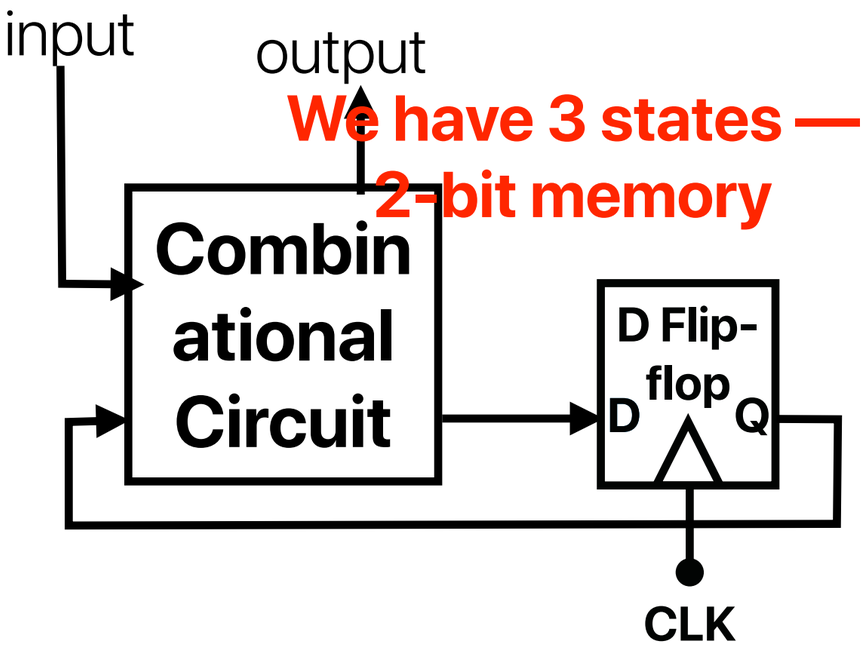


(A)



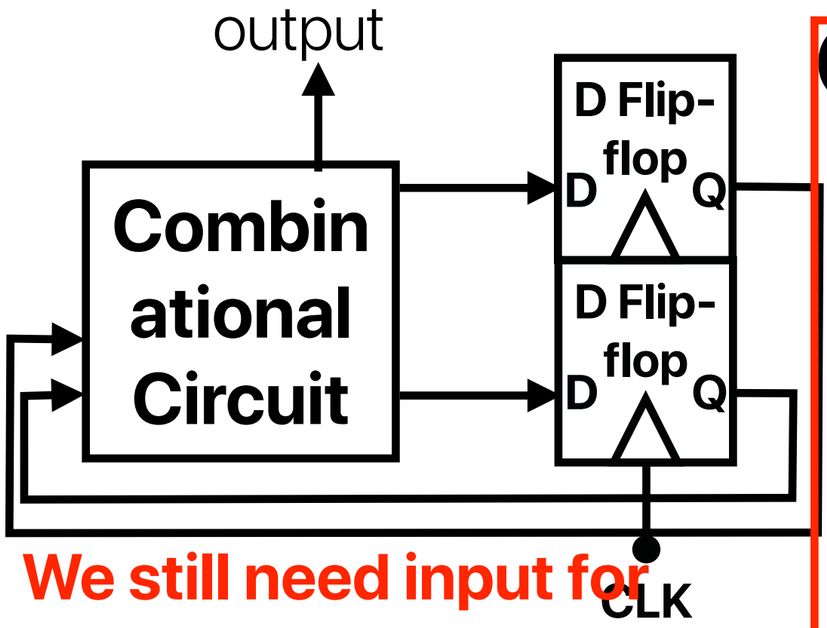
We need memory

(B)



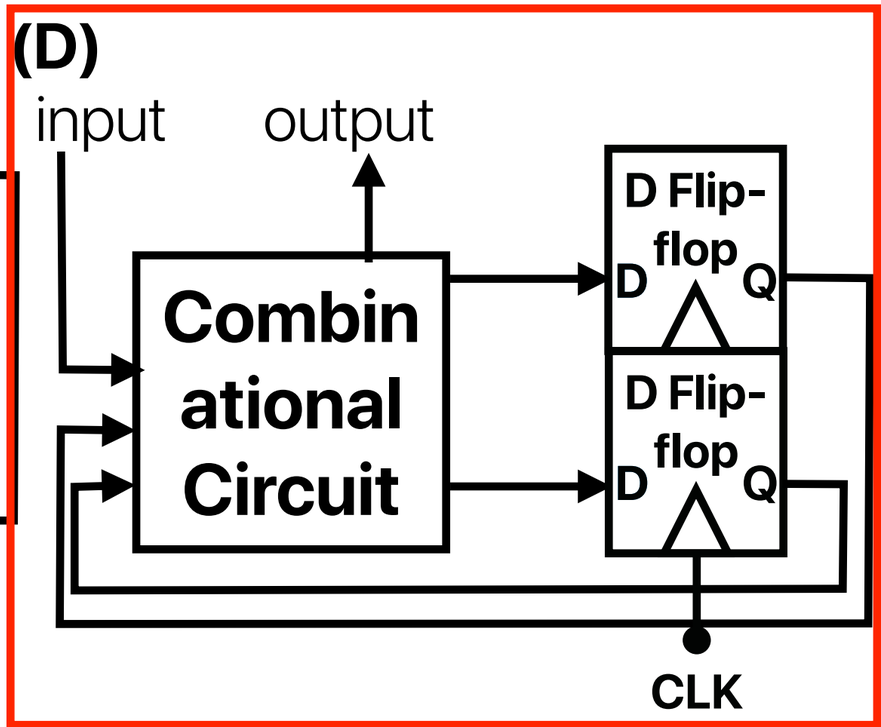
We have 3 states — 2-bit memory

(C)



We still need input for transition

(D)

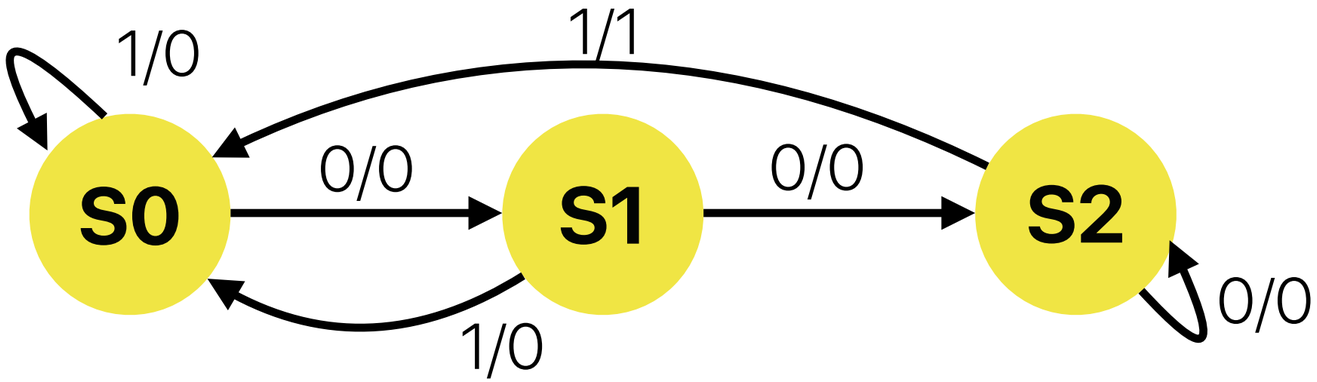


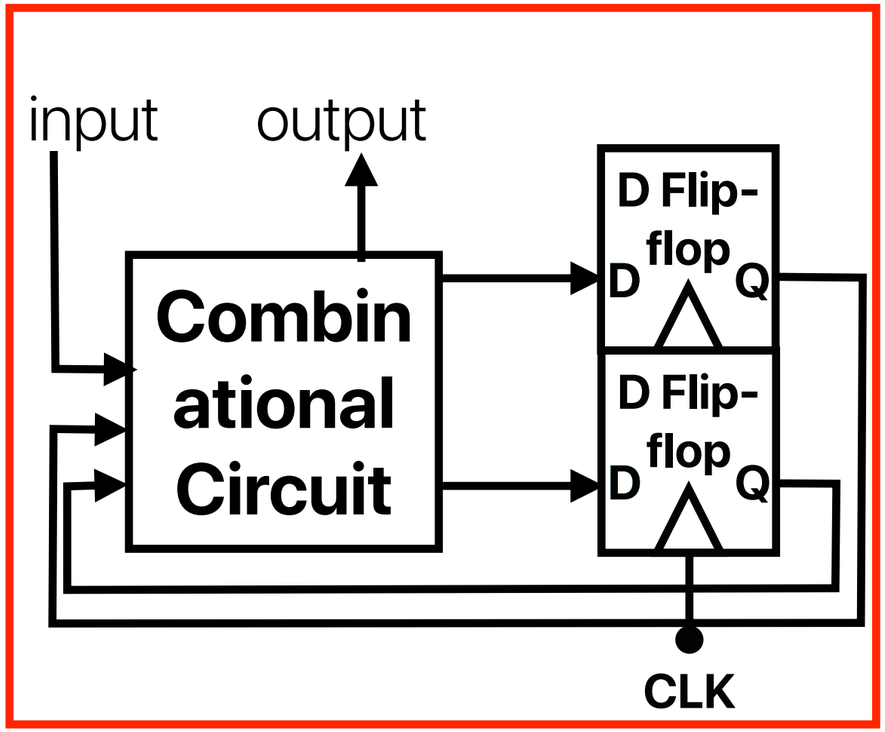
Sequential Circuit Design Flow

- Input Output Relation
- State Diagram (Transition of states)
  - State minimization (Reduction)
  - Finite state machine partitioning
- State Assignment (Map states into binary code)
  - Binary code, Gray encoding, One hot encoding, Coding optimization
- State Table (Truth table of states)
- Excitation Table (Truth table of FF inputs)
  - K Map, Minimal Expression
  - Logic Diagram

Life on Mars

State Diagram



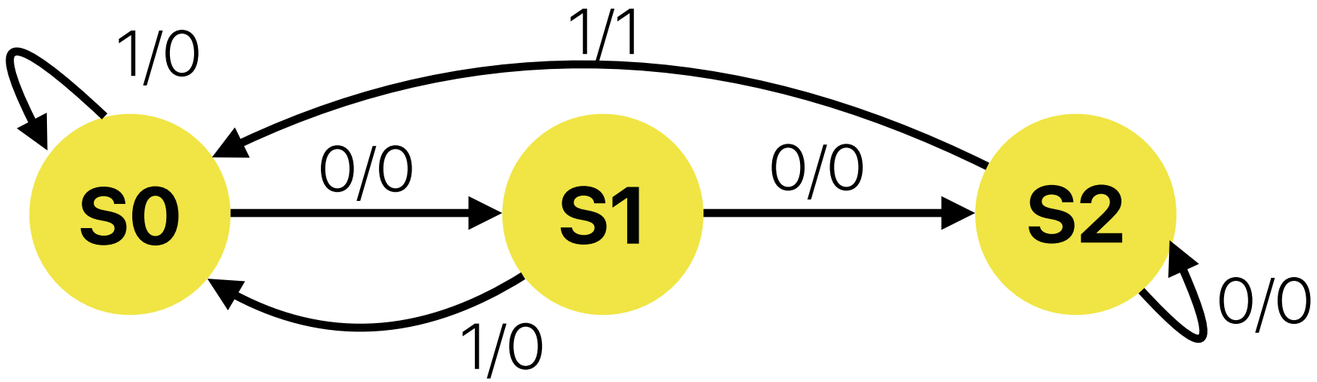


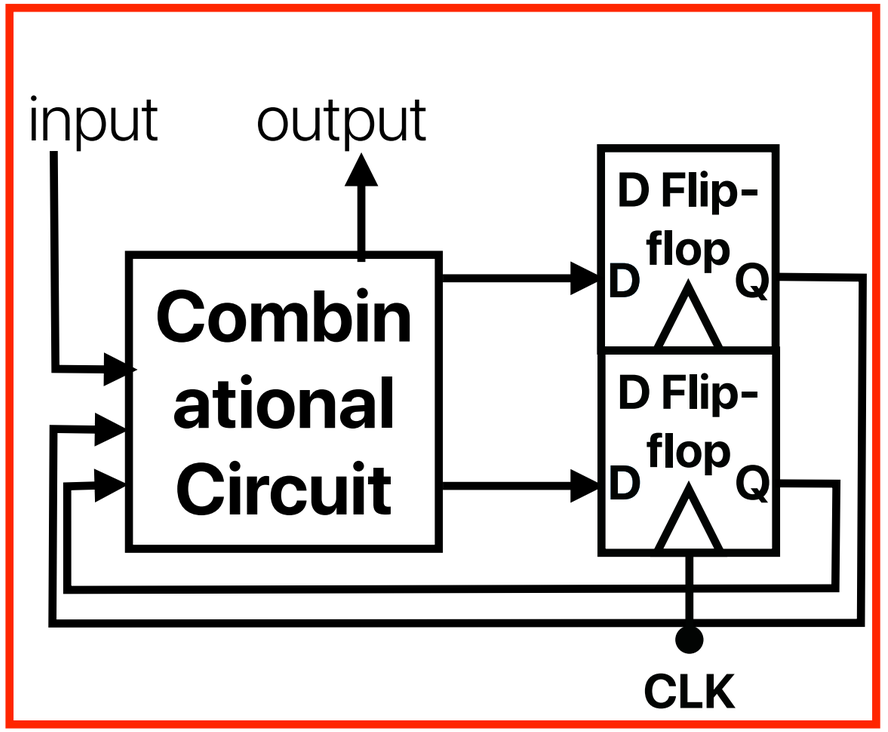
State Diagram

Current State	Next State, Output Input	
	0	1
S0	S1, 0	S0, 0
S1	S2, 0	S0, 0
S2	S2, 0	S0, 1

Life on Mars

State Diagram





State Diagram

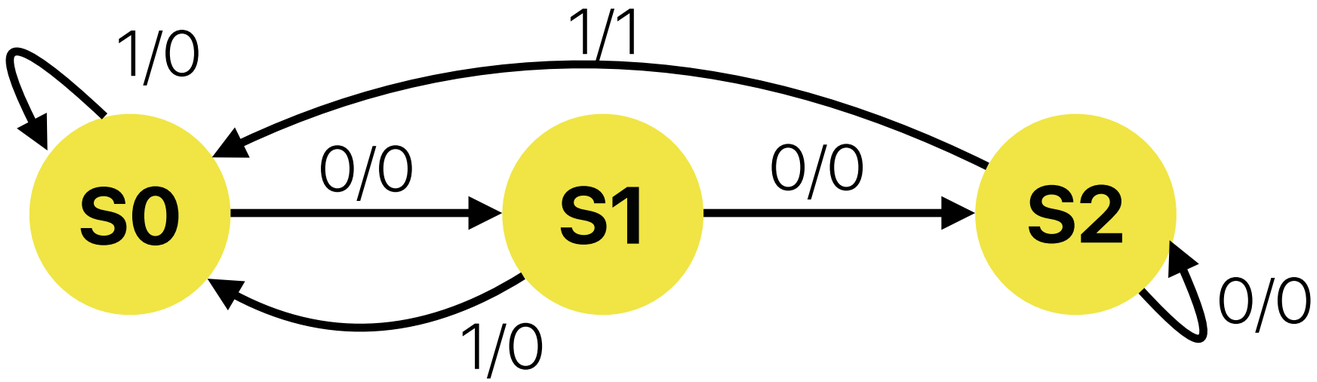
Current State	Next State, Output Input	
	0	1
S0	S1, 0	S0, 0
S1	S2, 0	S0, 0
S2	S2, 0	S0, 1

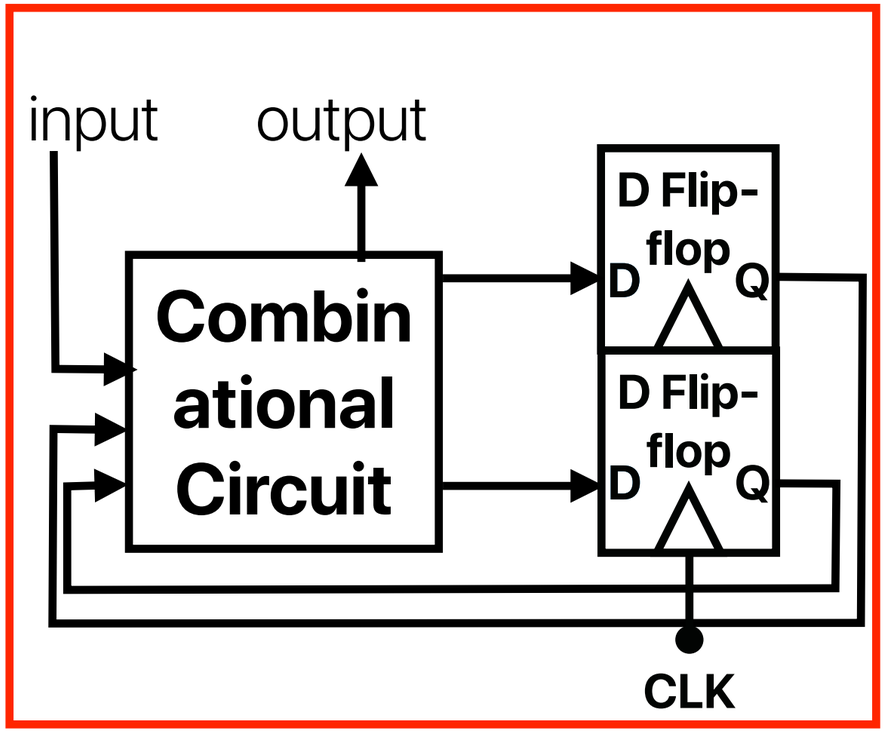
State Assignment

S0	00
S1	01
S2	10

Life on Mars

State Diagram





State Diagram

Current State	Next State, Output Input	
	0	1
S0	S1, 0	S0, 0
S1	S2, 0	S0, 0
S2	S2, 0	S0, 1

State Assignment

S0	00
S1	01
S2	10

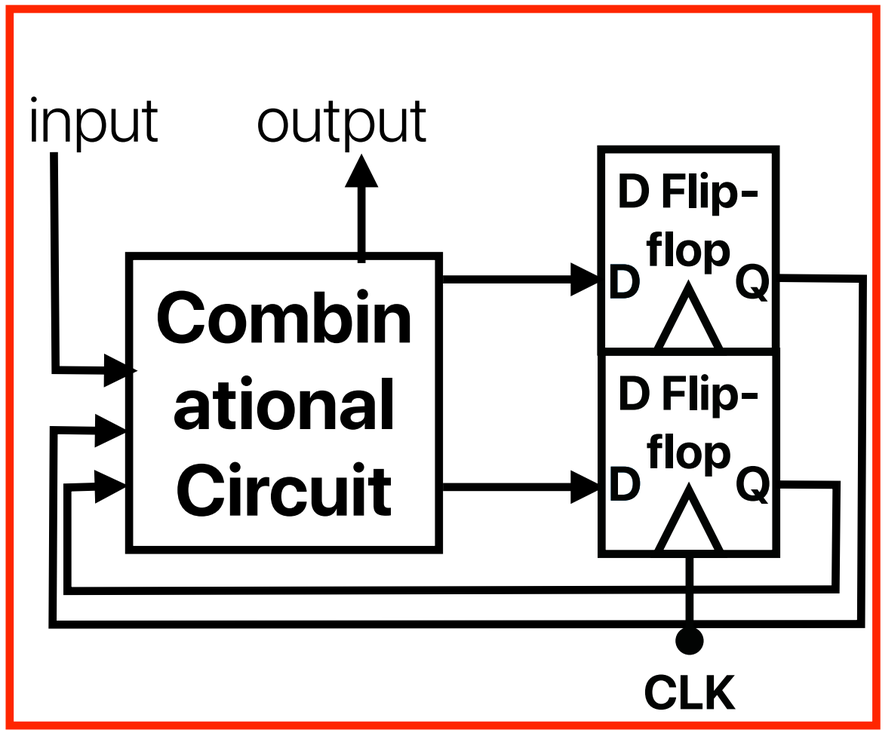
State Truth Table

State\Input	0	1
00	01, 0	00, 0
01	10, 0	00, 0
10	10, 0	00, 1

Poll close in 1:30

Excitation Table

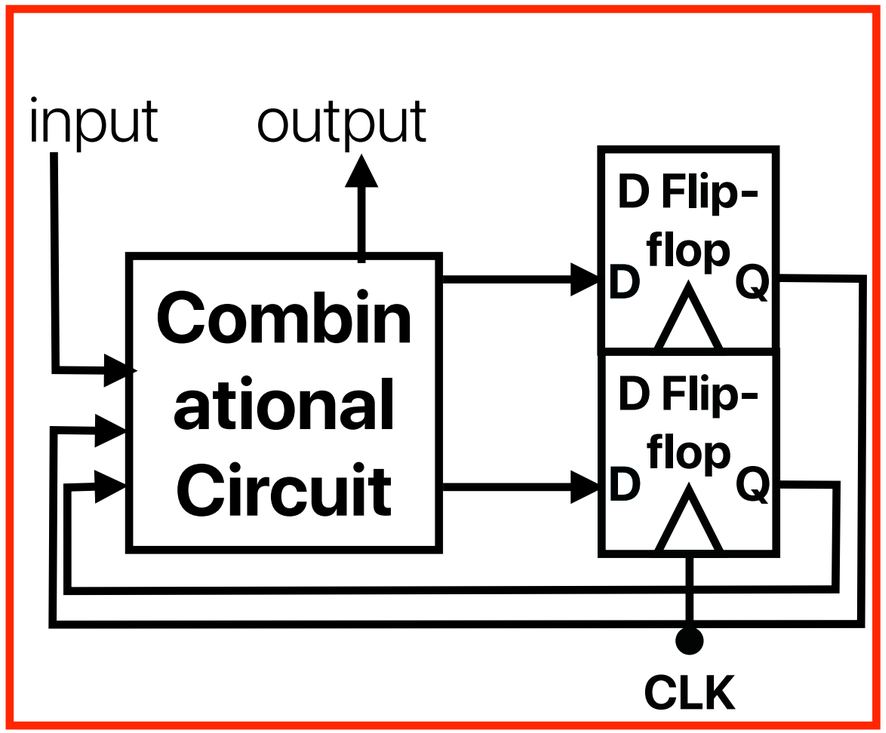
- Excitation table is basically the truth table describing the combinational circuit that provides inputs for the flip-flops in the sequential circuit. How many rows are there in the excitation table of Life on Mars?
  A. 2
  B. 3
  C. 4
  D. 8
  E. 32



Excitation Table

- Excitation table is basically the truth table describing the combinational circuit that provides inputs for the flip-flops in the sequential circuit. How many rows are there in the excitation table of Life on Mars?

A. 2
B. 3
C. 4
D. 8
E. 32



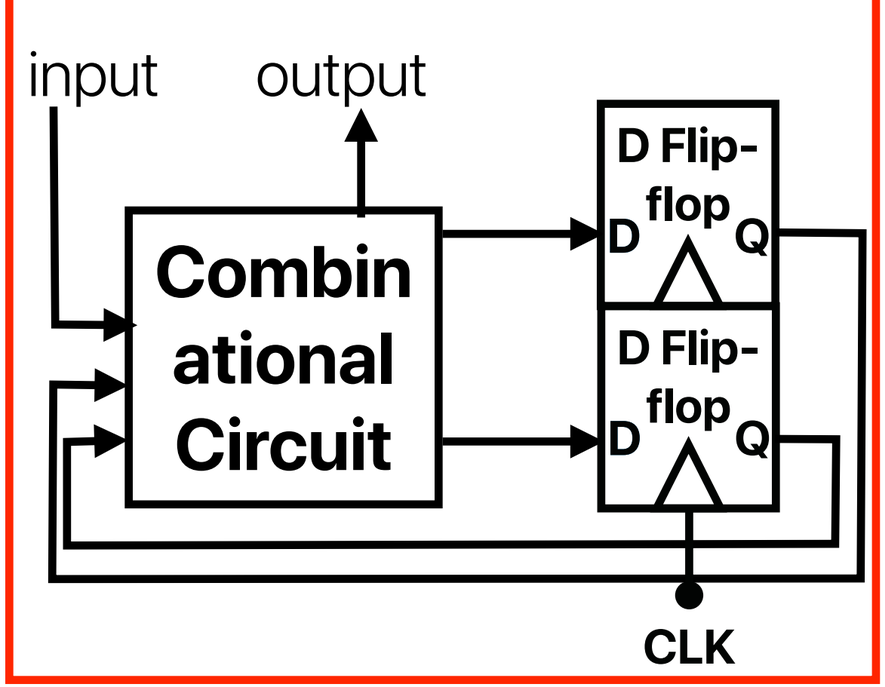
Life on Mars

State Truth Table

State\Input	0	1
00	01, 0	00, 0
01	10, 0	00, 0
10	10, 0	00, 1

Excitation Table

NextStateOput StateInput	D1	D0	y
000	0	1	0
001	0	0	0
010	1	0	0
011	0	0	0
100	1	0	0
101	0	0	1
110	X	X	X
111	X	X	X



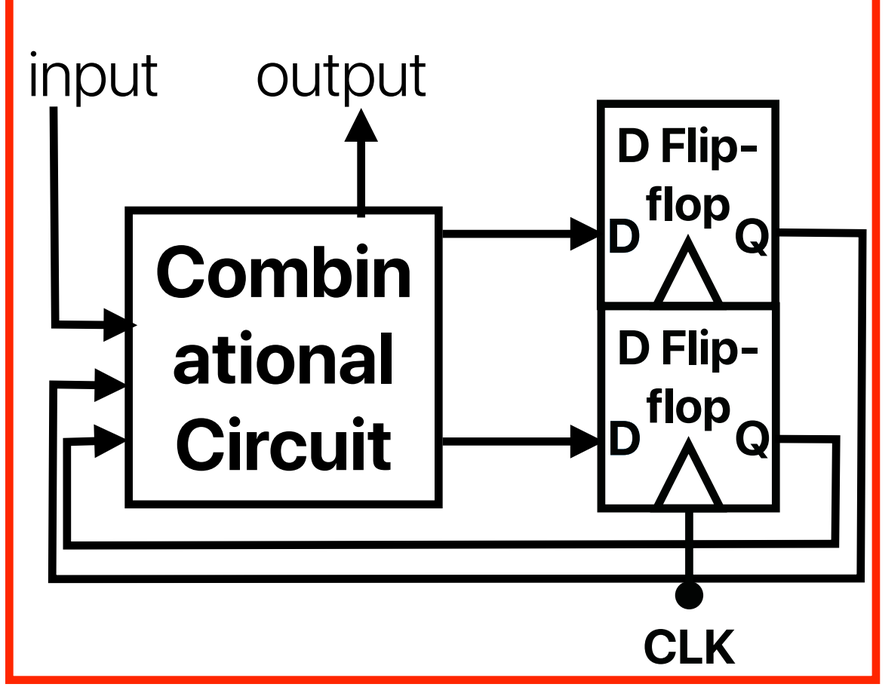
Life on Mars

State Truth Table

State\Input	0	1
00	01, 0	00, 0
01	10, 0	00, 0
10	10, 0	00, 1

Excitation Table

NextStateOput StateInput	D1	D0	y
000	0	1	0
001	0	0	0
010	1	0	0
011	0	0	0
100	1	0	0
101	0	0	1
110	X	X	X
111	X	X	X



K-Map — D1

	0,0	0,1	1,1	1,0
0	0	1	X	1
1	0	0	X	0

$D1 = x'Q_0 + x'Q_1$

K-Map — D0

	0,0	0,1	1,1	1,0
0	1	0	X	0
1	0	0	X	0

$D0 = Q_0'Q_1'x'$

K-Map — y

	0,0	0,1	1,1	1,0
0	0	0	X	0
1	0	0	X	1

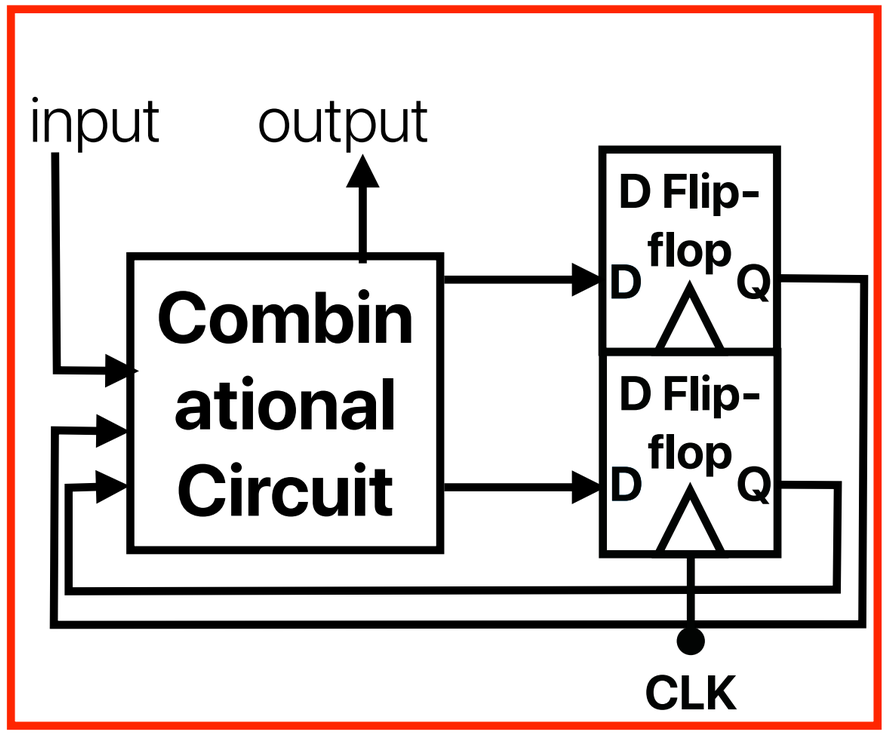
$y = Q_1'x$

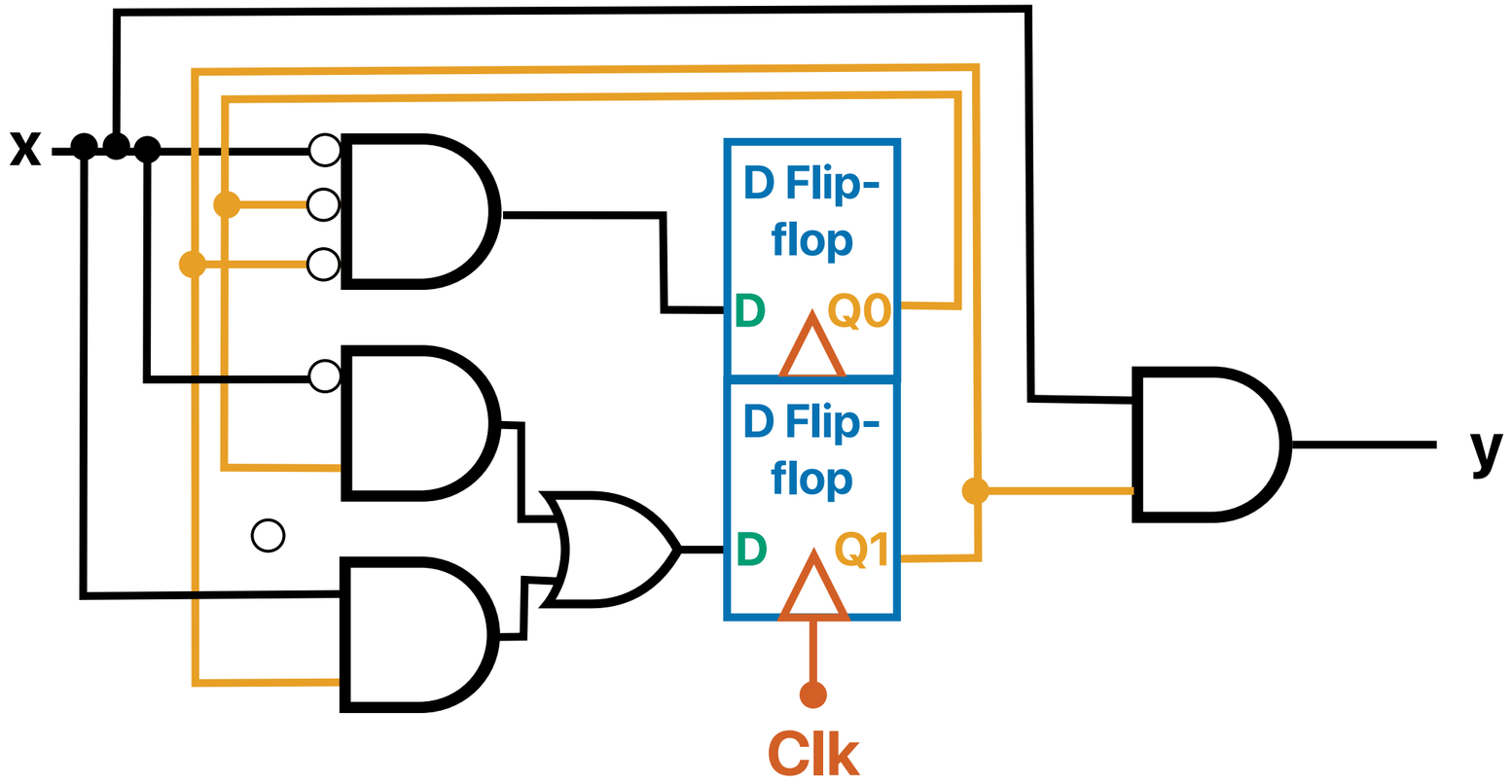
Circuit — Life on Mars

$D1 = x'Q_0 + x'Q_1$

$D0 = Q_0'Q_1'x'$

$y = Q_1'x$

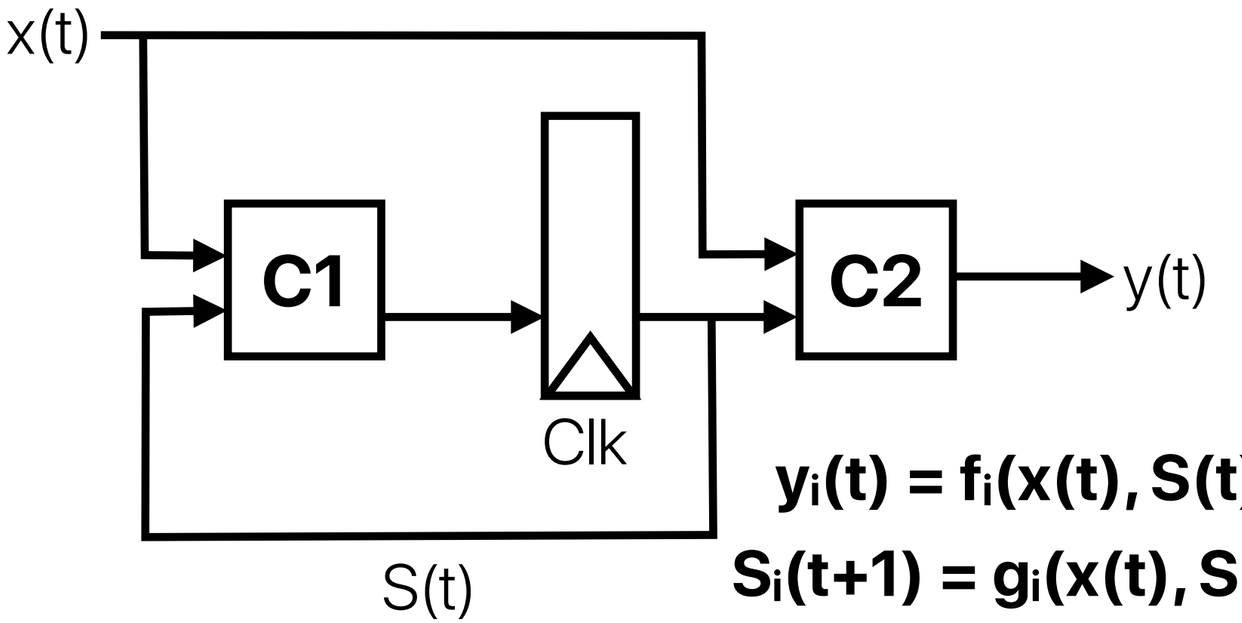




Canonical Form: Mealy and Moore Machines

Canonical Form: Mealy and Moore Machines

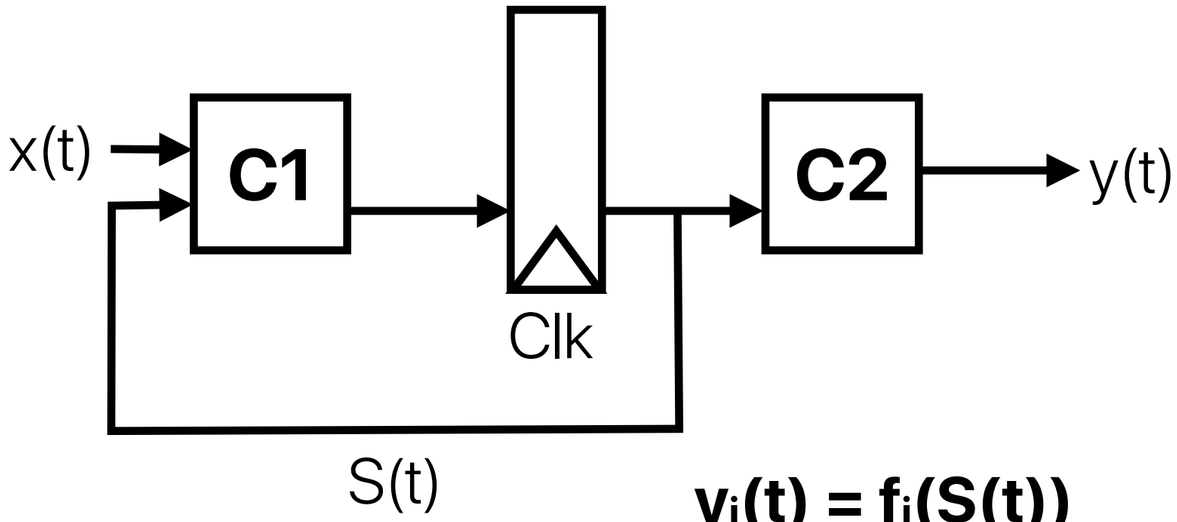
Mealy Machine



$$y_i(t) = f_i(x(t), S(t))$$

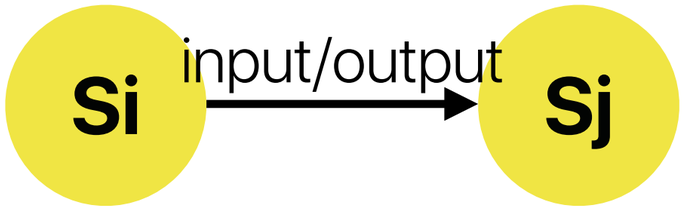
$$S_i(t+1) = g_i(x(t), S(t))$$

Moore Machine

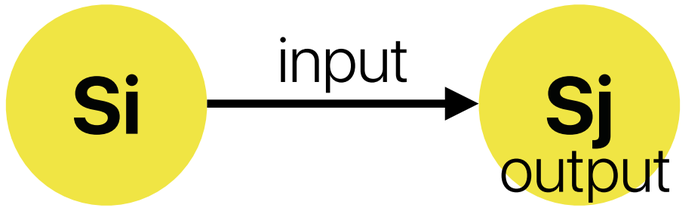


$$y_i(t) = f_i(S(t))$$

$$S_i(t+1) = g_i(x(t), S(t))$$



Result depends on both input and the current state

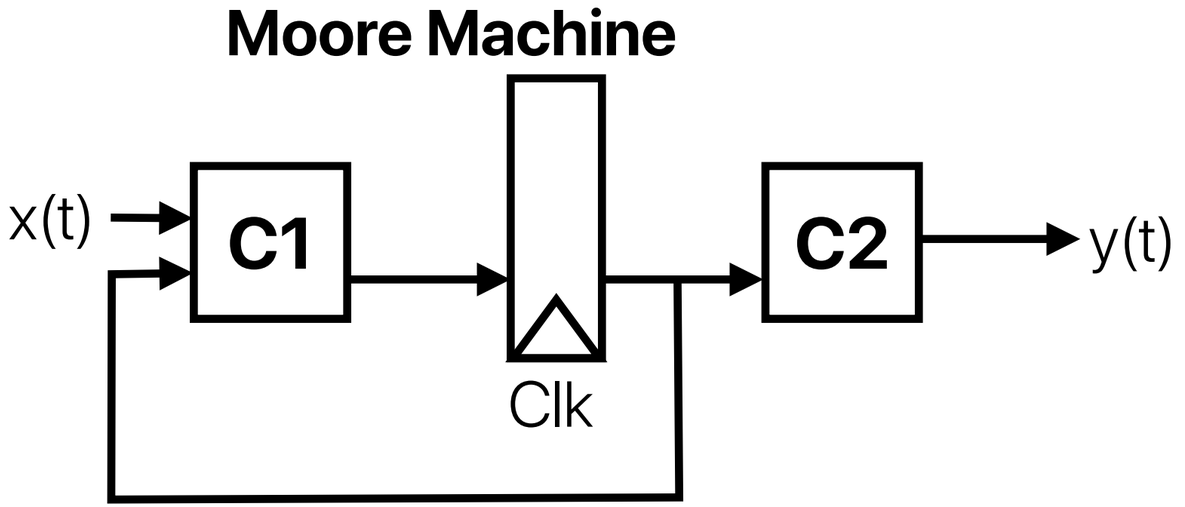


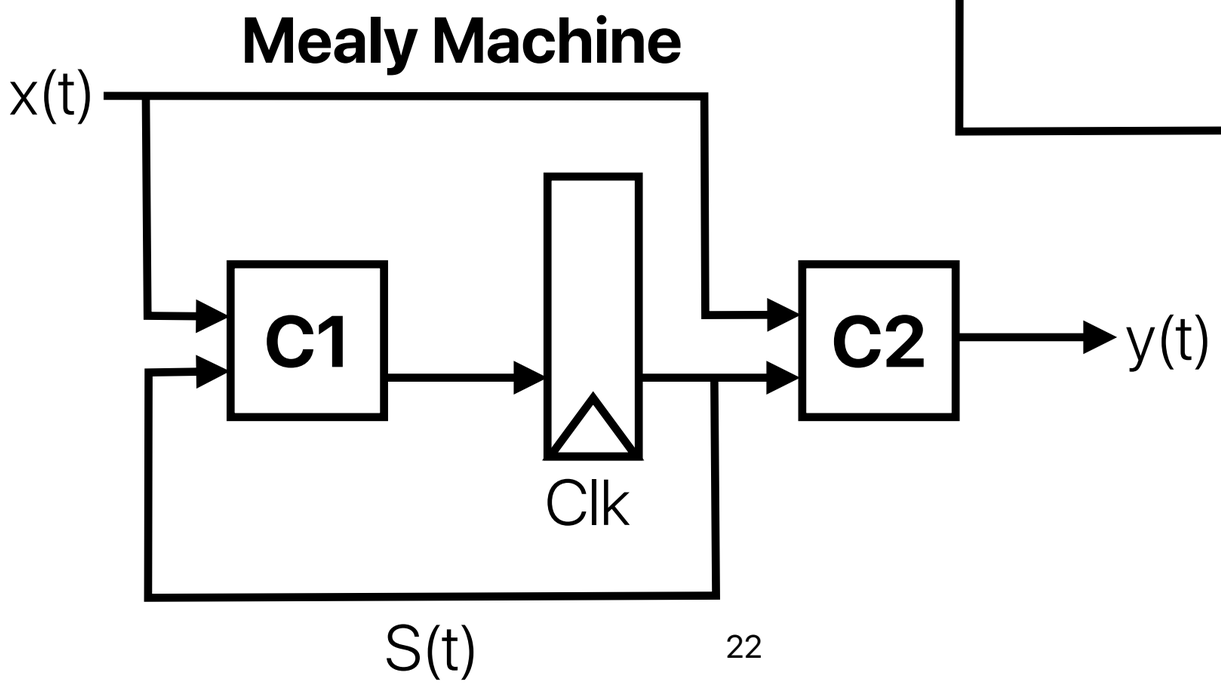
Result only depends on the current state

Poll close in 1:30

Moore or Mealy? — Life on Mars

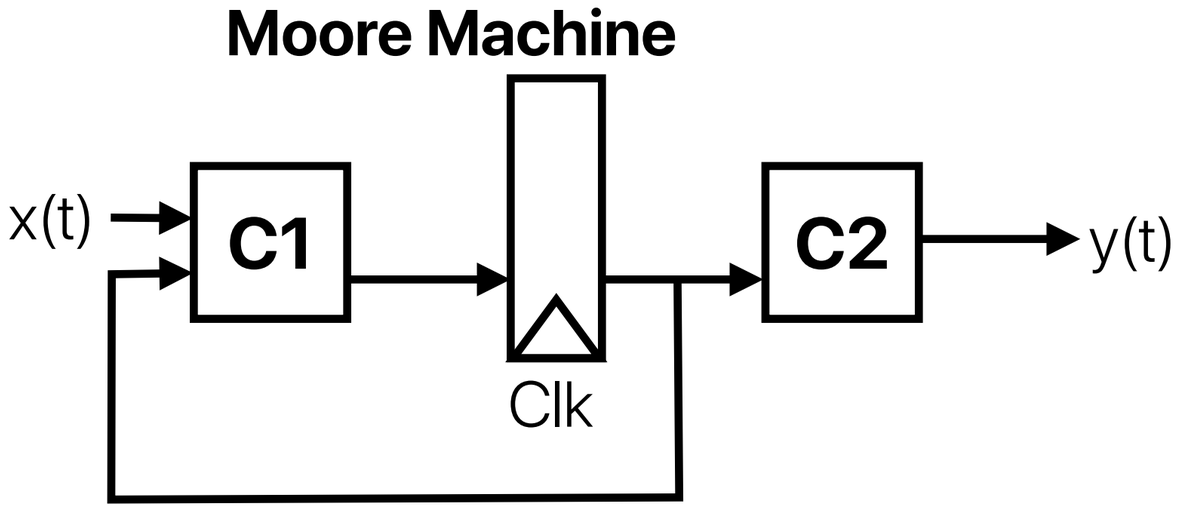
- Which type of state machine can describe the "Life on Mars" pattern recognizer of "001".
  A. Moore machine
  B. Mealy machine
  C. Both
  D. None

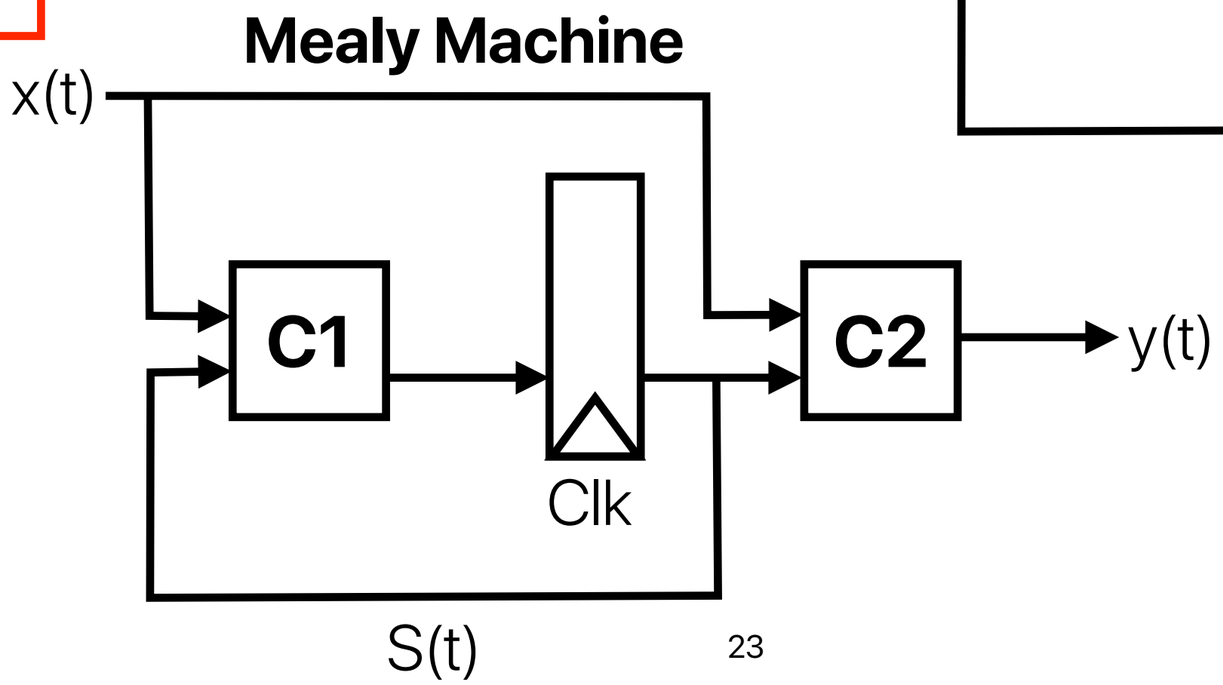




Moore or Mealy? — Life on Mars

- Which type of state machine can describe the "Life on Mars" pattern recognizer of "001".
  A. Moore machine
  B. Mealy machine
  C. Both
  D. None

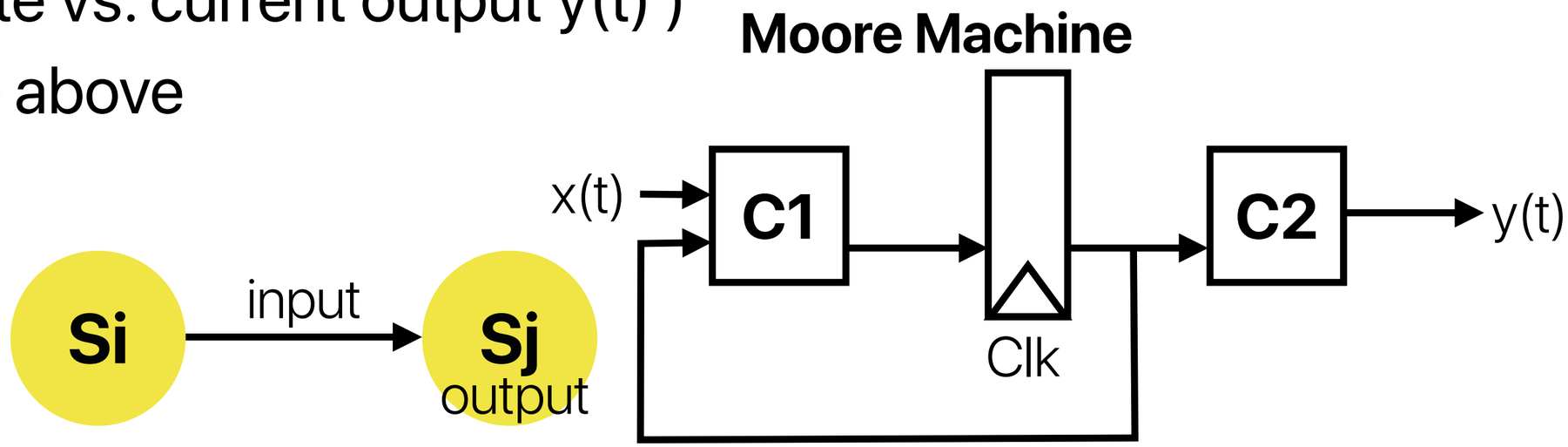




Poll close in 1:30

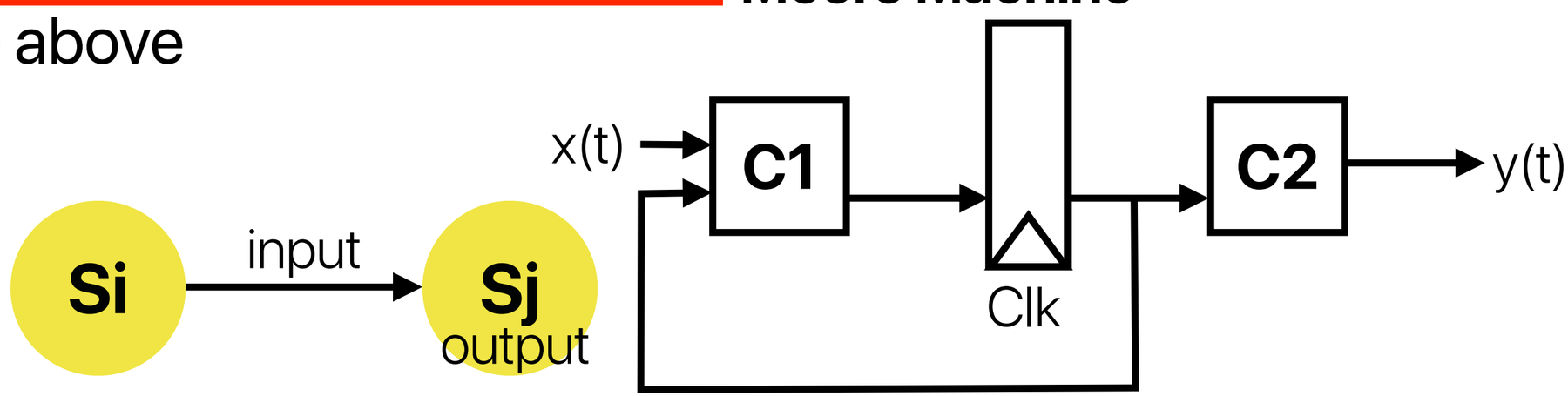
Moore machine for Life on Mars

- What does state table need to show to design controls of C2 to implement pattern recognizer "001"?
  A. (current input x(t), current state S(t) vs. next state, S(t+1))
  B. (current input, current state vs. current output y(t))
  C. (current state vs. current output y(t) and next state)
  D. (current state vs. current output y(t))
  E. None of the above



Moore machine for Life on Mars

- What does state table need to show to design controls of C2 to implement pattern recognizer "001"?
  A. (current input x(t), current state S(t) vs. next state, S(t+1))
  B. (current input, current state vs. current output y(t))
  C. (current state vs. current output y(t) and next state)
  D. (current state vs. current output y(t))
  E. None of the above



Conversion from Mealy to Moore

① Identify distinct (Next State, y) pair

② Replace each distinct (Next State, y) pair with distinct new states

③ Insert rows of present state = new states

④ Append each present state with its output y

Current State	Next State, Input 0	Next State, Input 1
S0	S1, 0	S0, 0
S1	S2, 0	S0, 0
S2	S2, 0	**S0, 1**

Current State	Next State, Input 0	Next State, Input 1	Output
S0	S1	S0	0
S1	S2	S0	0
S2	S2	**S3**	0
S3			

Poll close in 1:30

Conversion from Mealy to Moore

① Identify distinct (Next State, y) pair

② Replace each distinct (Next State, y) pair with distinct new states

③ Insert rows of present state = new states

④ Append each present state with its output y

- For the given Moore machine, what are the next states with respect to present state S3?
  - A. S2, S3, 1
  - B. S2, S0, 1
  - C. S1, S0, 1
  - D. S1, S0. 0
  - E. None of the above.

Current State	Next State Input 0	Next State Input 1
S0	S1, 0	S0, 0
S1	S2, 0	S0, 0
S2	S2, 0	**S0, 1**

Current State	Next State Input 0	Next State Input 1	Output
S0	S1	S0	0
S1	S2	S0	0
S2	S2	**S3**	0
S3			

Conversion from Mealy to Moore

① Identify distinct (Next State, y) pair

② Replace each distinct (Next State, y) pair with distinct new states

③ Insert rows of present state = new states

④ Append each present state with its output y

- For the given Moore machine, what are the next states with respect to present state S3?
  - A. S2, S3, 1
  - B. S2, S0, 1
  - C. S1, S0, 1
  - D. S1, S0, 0
  - E. None of the above.

Current State	Next State Input 0	Next State Input 1
S0	S1, 0	S0, 0
S1	S2, 0	S0, 0
S2	S2, 0	**S0, 1**

Current State	Next State Input 0	Next State Input 1	Output
S0	S1	S0	0
S1	S2	S0	0
S2	S2	**S3**	0
S3	**S1**	**S0**	**1**

Conversion from Mealy to Moore

① Identify distinct (Next State, y) pair

② Replace each distinct (Next State, y) pair with distinct new states

③ Insert rows of present state = new states

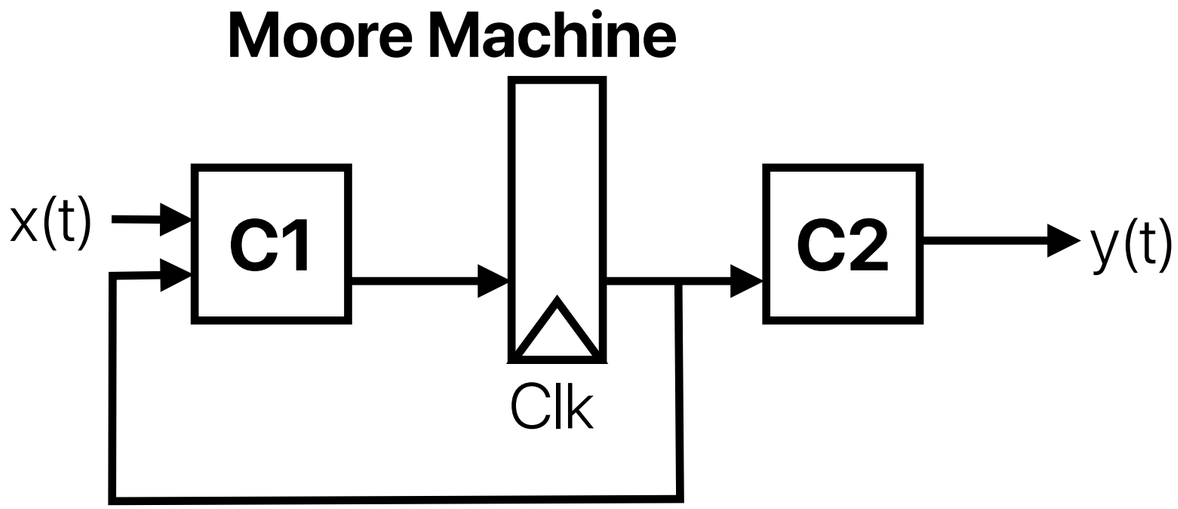
④ Append each present state with its output y

Current State	Next State, Input 0	Next State, Input 1
S0	S1, 0	S0, 0
S1	S2, 0	S0, 0
S2	S2, 0	**S0, 1**

Current State	Next State, Input 0	Next State, Input 1	Output
S0	S1	S0	0
S1	S2	S0	0
S2	S2	**S3**	0
S3	**S1**	**S0**	**1**

State tables for C1 and C2

Current State		Next State Input 0	Next State Input 1	Output
S0	00	S1	S0	0
S1	01	S2	S0	0
S2	10	S2	S3	0
S3	11	S1	S0	1



C1

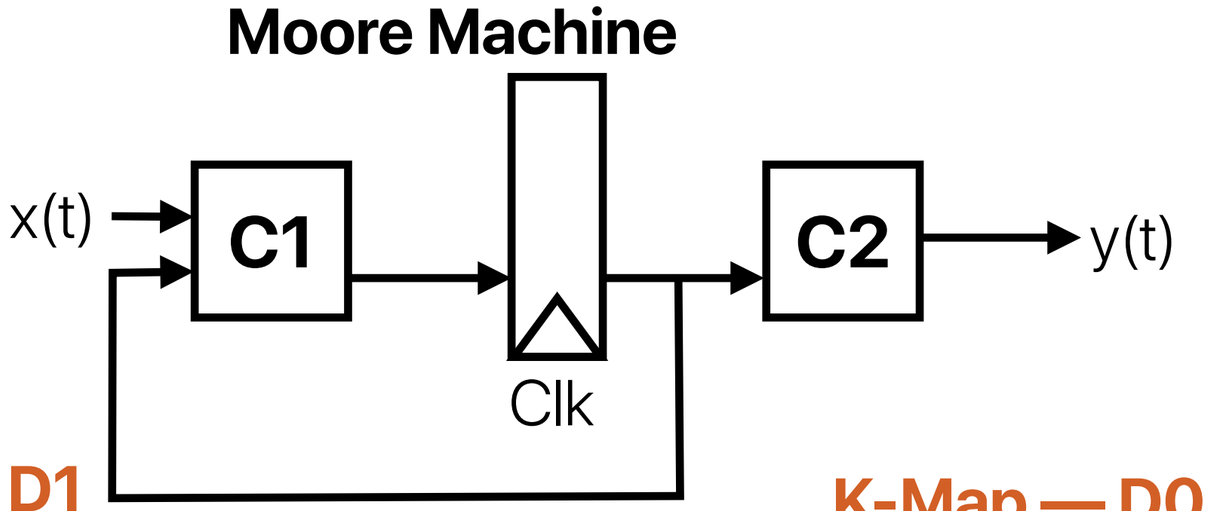
CurrentState-Input	D1	D0
000	0	1
001	0	0
010	1	0
011	0	0
100	1	0
101	1	1
110	0	1
111	0	0

C2

State	y
00	0
01	0
10	0
11	1

State tables for C1 and C2

Current State		Next State Input 0	Next State Input 1	Output
S0	00	S1	S0	0
S1	01	S2	S0	0
S2	10	S2	S3	0
S3	11	S1	S0	1



C1

CurrentState-Input	D1	D0
000	0	1
001	0	0
010	1	0
011	0	0
100	1	0
101	0	0
110	0	1
111	0	0

C2

State	y
00	0
01	0
10	0
11	1

K-Map — D1

	0,0	0,1	1,1	1,0
0	0	1	0	1
1	0	0	0	1

$D1 = x'Q_1'Q_0 + Q_1Q_0'$

K-Map — D0

	0,0	0,1	1,1	1,0
0	1	0	1	0
1	0	0	0	1

$D0 = Q_0'Q_1'x' + Q_0Q_1x' + Q_0'Q_1x'$

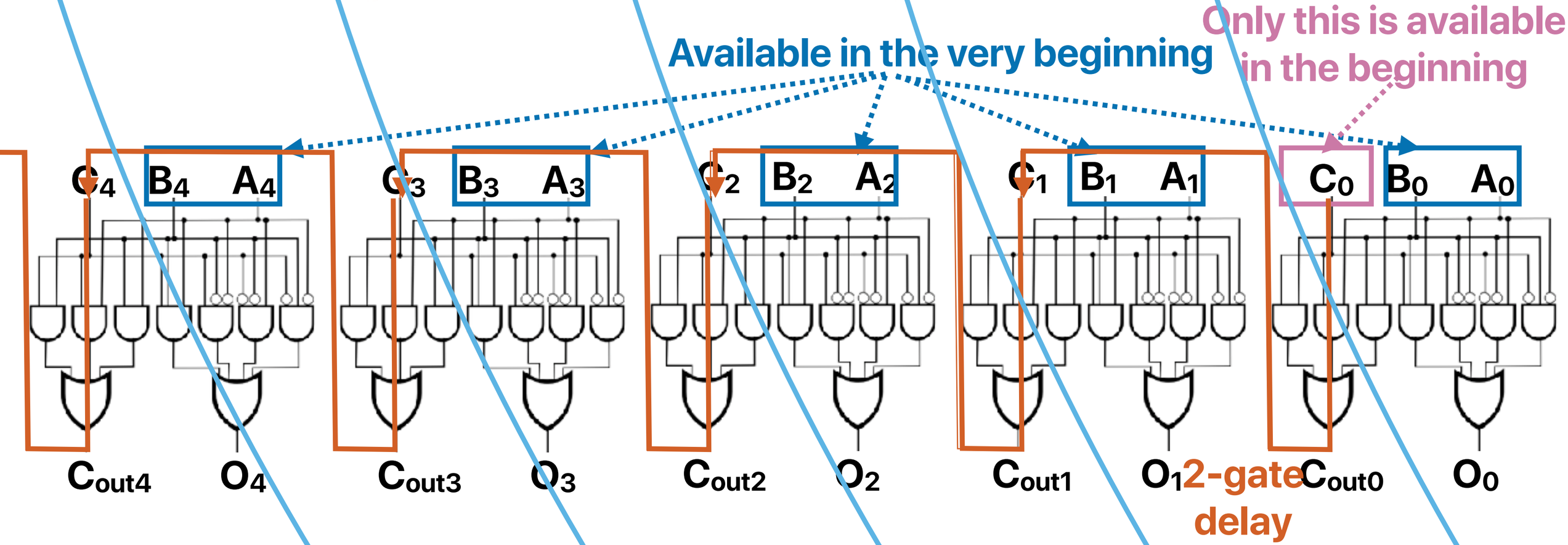
K-Map — y

	0	1
0	0	0
1	0	1

$y = Q_0Q_1$

When sequential circuits meet datapath components (3)

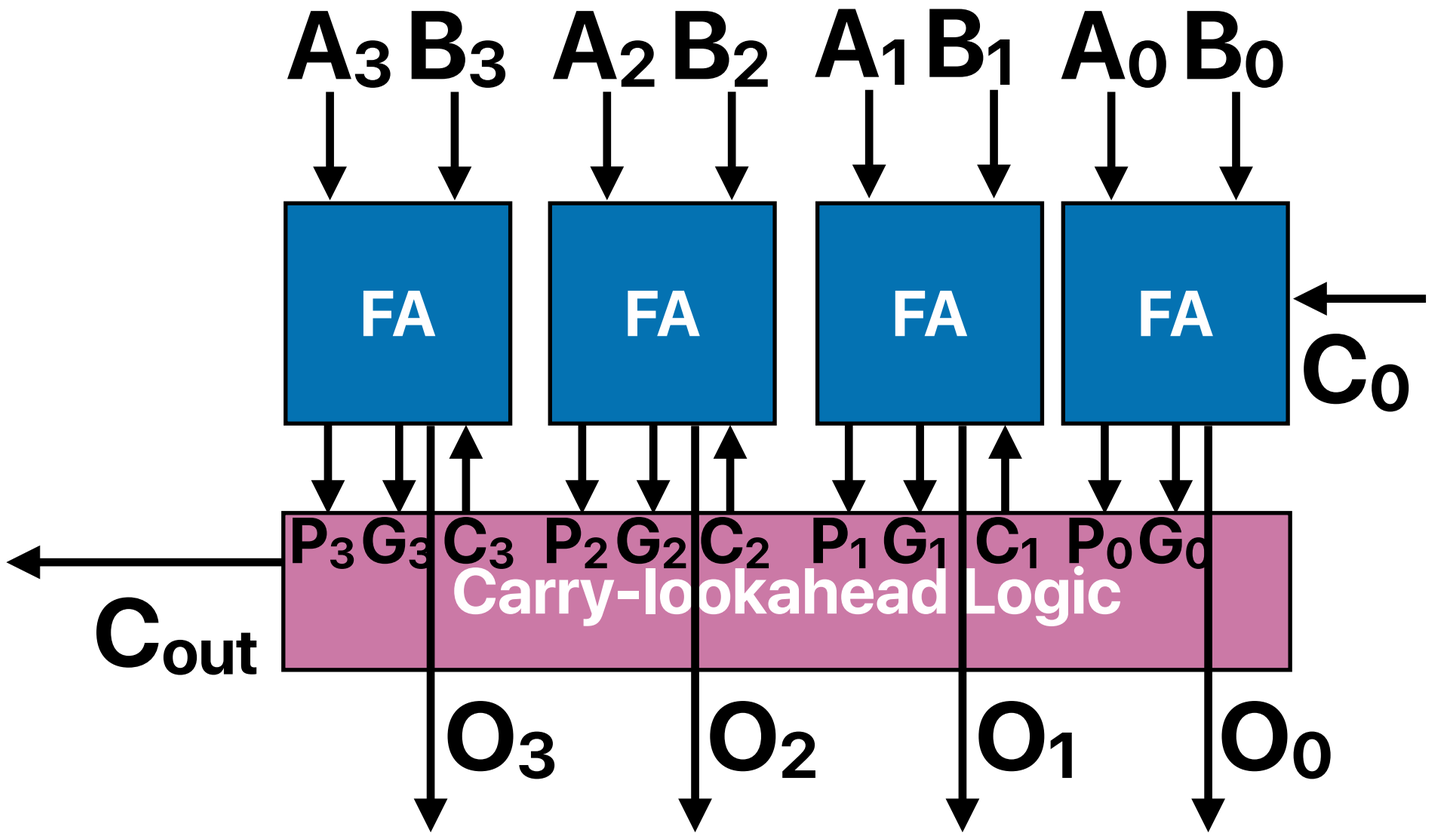
The delay is determined by the "critical path"



Carry-Ripple Adder

CLA (cont.)

- All "G" and "P" are immediately available (only need to look over Ai and Bi), but "c" are not (except the c0).



$$G_i = A_i B_i$$

$$P_i = A_i \text{ XOR } B_i$$

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0)$$
$$= G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2$$
$$= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

$$C_4 = G_3 + P_3 C_3$$
$$= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$$

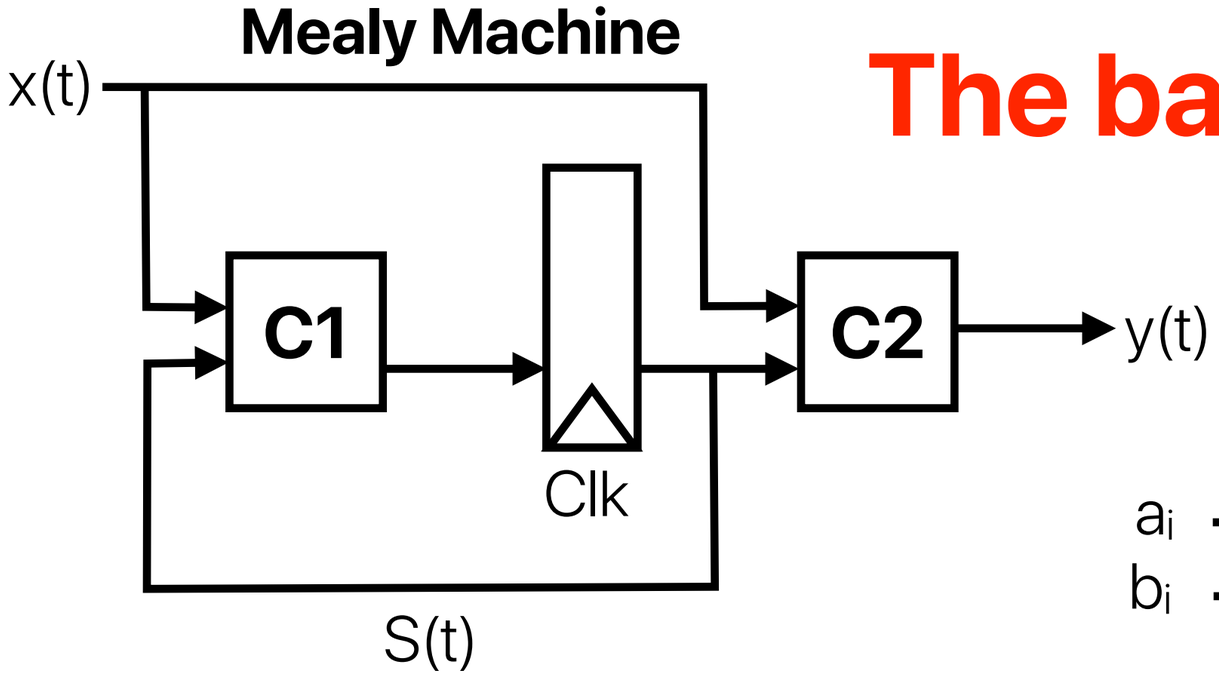
CLA v.s. Carry-ripple

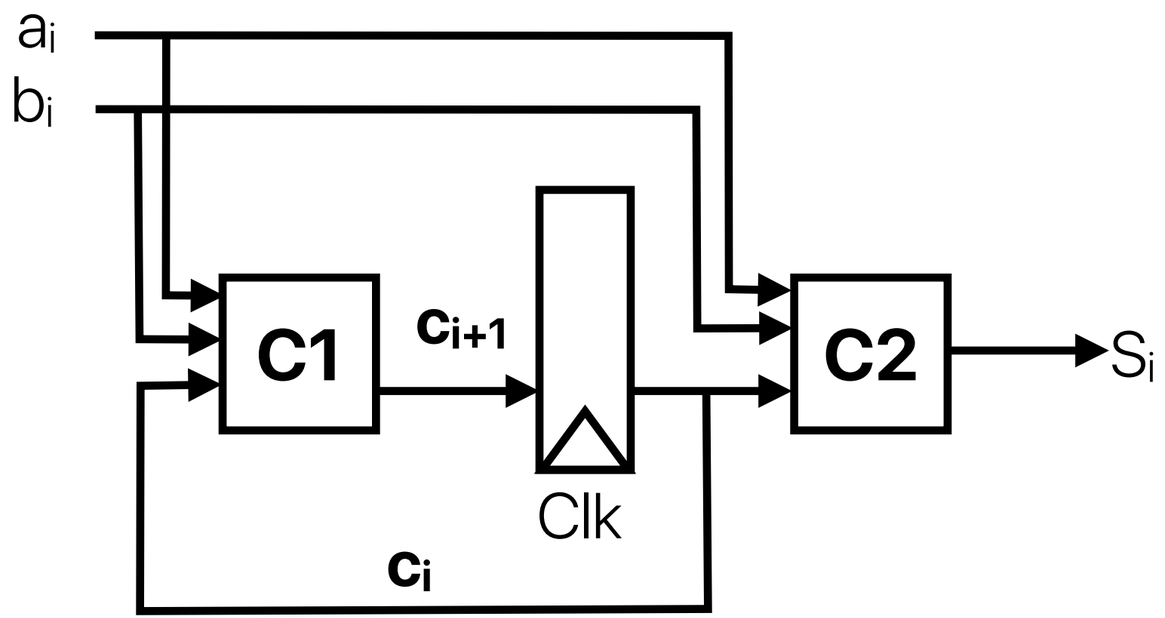
- Size:
  - 32-bit CLA with 4-bit CLAs — requires 8 of 4-bit CLA
    - Each requires 116 for the CLA 4*(4*6+8) for the A+B — 244 gates
    - 1952 transistors
  - 32-bit CRA
    - 1600 transistors **Win!**
- Delay
  - 32-bit CLA with 8 4-bit CLAs
    - 2 gates * 8 = 16 **Win!**
  - 32-bit CRA
    - 64 gates

Area-Delay Trade-off!

Serial Adder

The basic idea





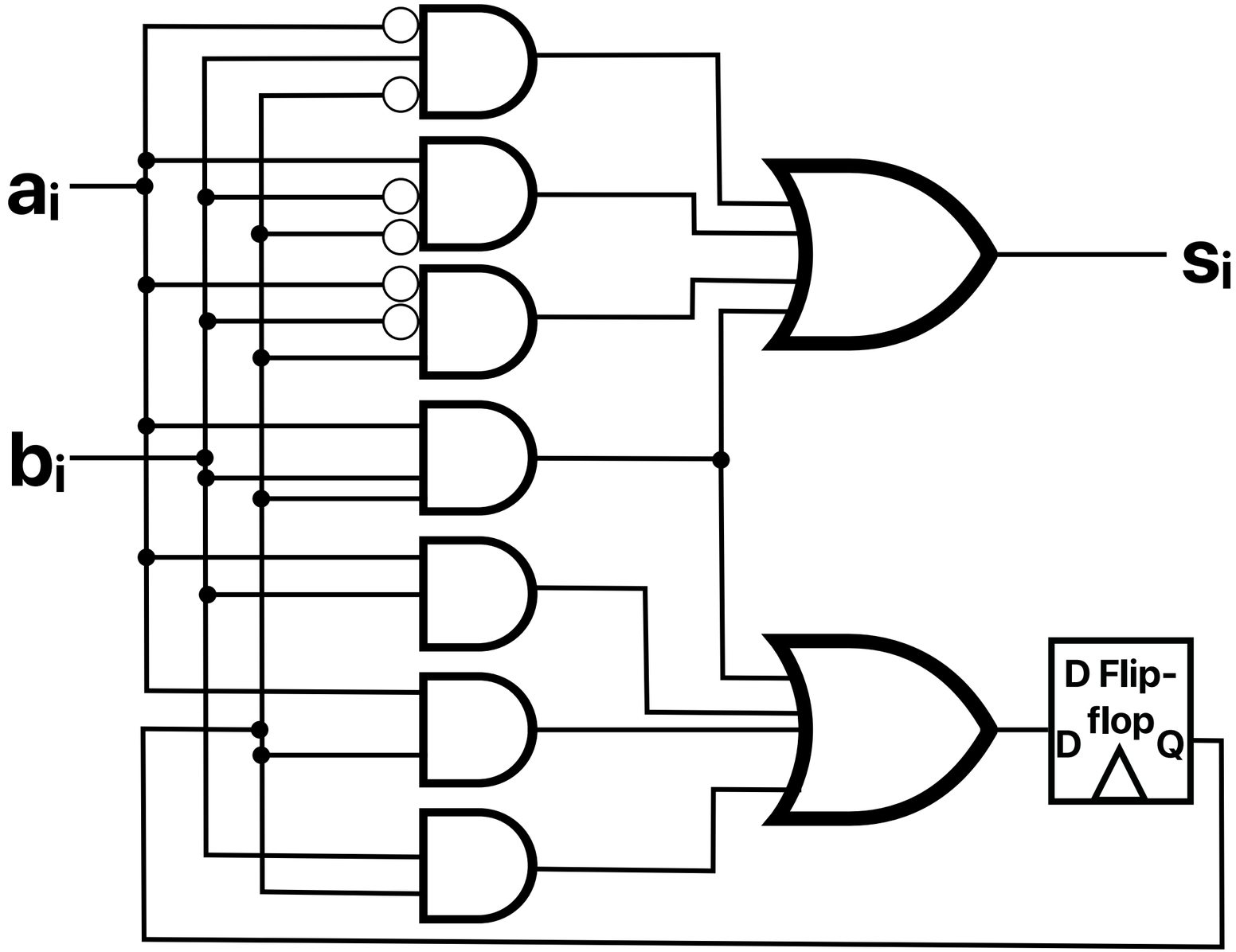
Feed a_i and b_i and generate s_i at time i. Where is c_i and c_{i+1}?

Excitation Table of Serial Adder

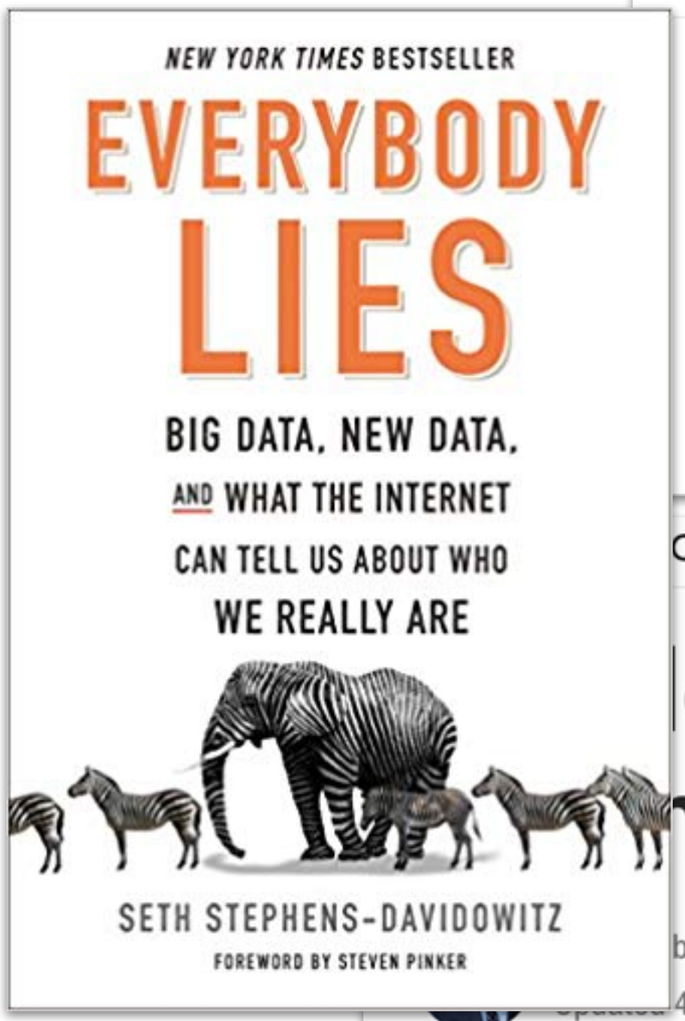
a_i	b_i	c_i	c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Excitation Table of Serial Adder

a_i	b_i	c_i	c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1




EXCUSE



Google

How Often Do People Lie

All News Images Videos Shopping More Settings Tools

About 267,000,000 results (0.54 seconds)

The study, published in the journal's June issue, found that 60 percent of **people lied** at least once during a 10-minute conversation and told an average of two to three lies. "**People** tell a considerable number of lies in everyday conversation. Jun 10, 2002

UMass researcher finds most people lie in everyday ...
https://www.eurekalert.org › pub_releases › uoma-urf061002

olitics 45 Congress SCOTUS Facts First 2020 2019 Elections LIVE TV Edition

ld Trump lies more often than you wash nands every day

by Chris Cillizza, CNN Editor-at-large

4:56 PM ET, Mon June 10, 2019

"A lie doesn't become truth, wrong doesn't become right and evil doesn't become good, just because it is accepted by a majority."

–Rick Warren

Honesty is the best policy

–Benjamin Franklin

Better three hours too soon, than one minute too late.

–William Shakespeare

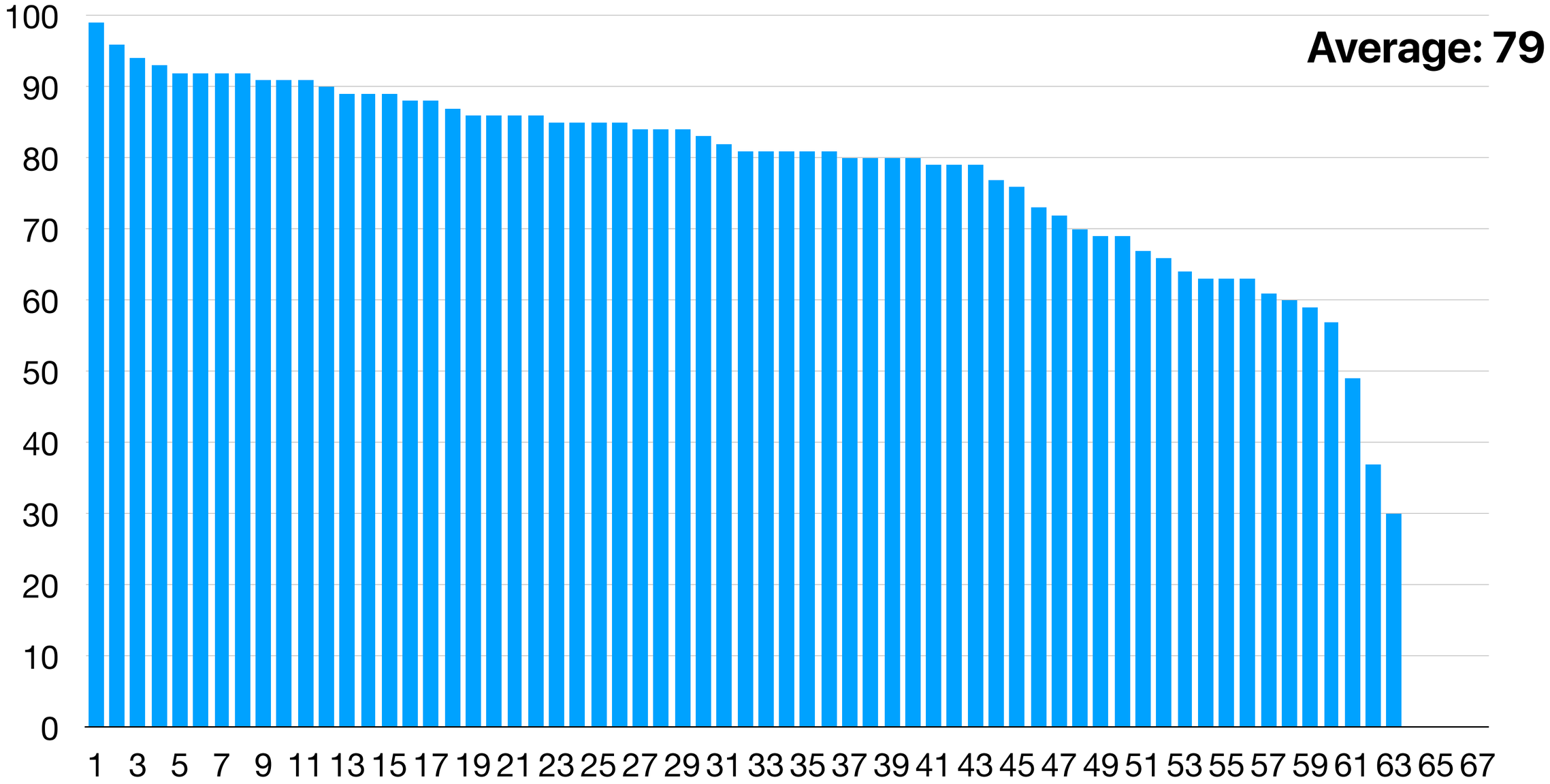
In truth, people can generally make time for what they choose to do; it is not really the time but the will that is lacking.

–Sir John Lubbock

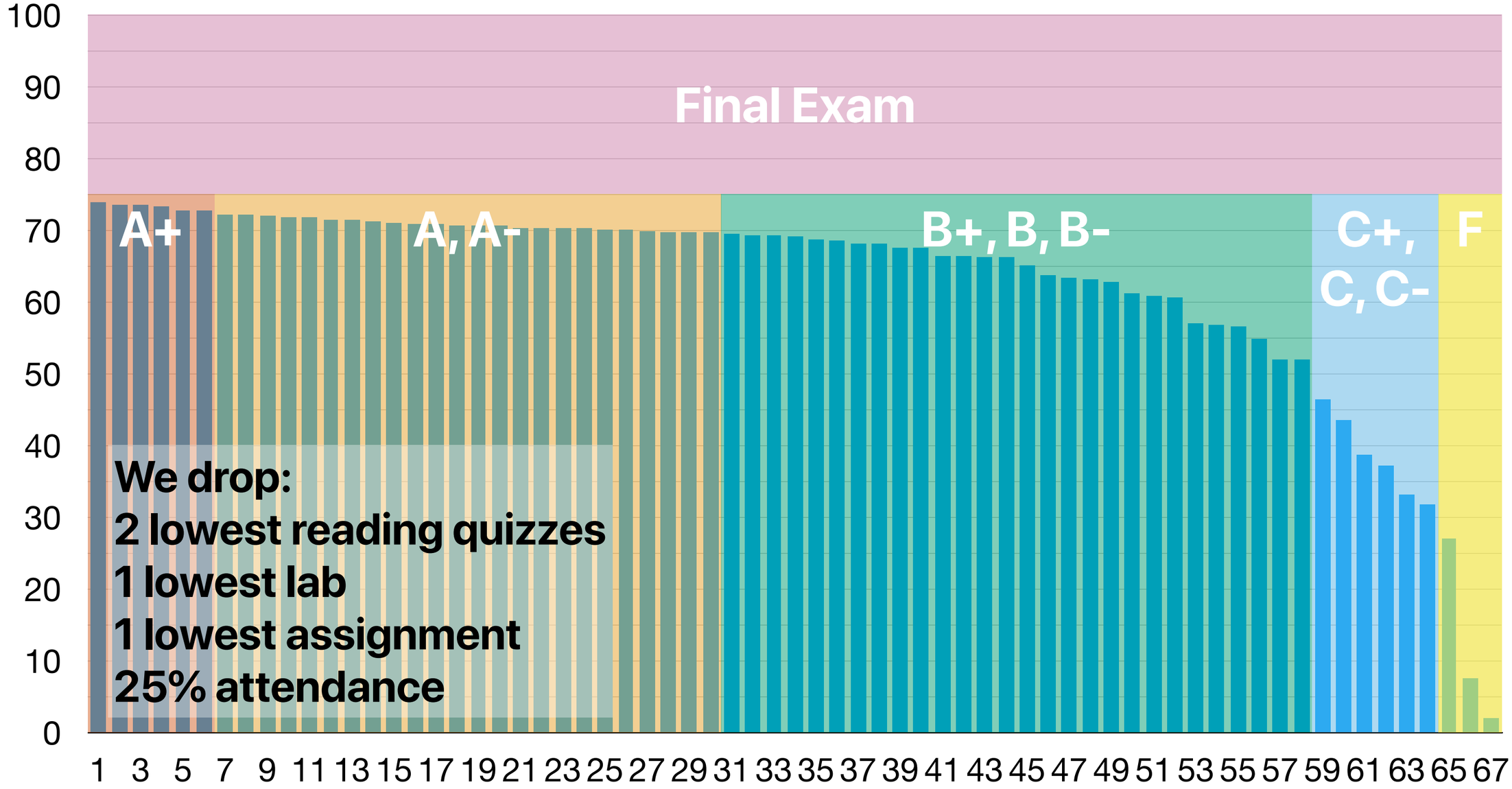
Don’t over-commit

–Prof. Usagi

Midterm


Average: 79
100
90
80
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67

Weighted Total — what really decides your final grades



Announcement

- Lab 4 — due tonight
- Reading quiz due this Thursday
- Assignment #4 due next Tuesday — **Chapter 4.8-4.9 & 5.2-5.4**
- Lab 5 is up — due next Thursday
  - Start early & plan your time carefully
  - Watch the video and read the instruction BEFORE your session
  - There are links on both course webpage and iLearn lab section
  - Submit through iLearn > Labs
- Check your grades in iLearn

Electrical
Computer Science
Engineering

120A

つづく