Computer Science 210 s1c

# Computer Systems 1

2013 Semester 1

Lecture Notes

Lecture 9, 22Mar13:

# Finite State Machines & the von Neumann Model

*James Goodman*


Credits: Slides prepared by Gregory T. Byrd, North Carolina State University

# Announcements

Facebook:

https://www.facebook.com/groups/324132367708838/#!/groups/324132367708838/

*Thanks to Gabby Freeland & Chris Petty!*

# Finite State Machine

A description of a system with the following components:

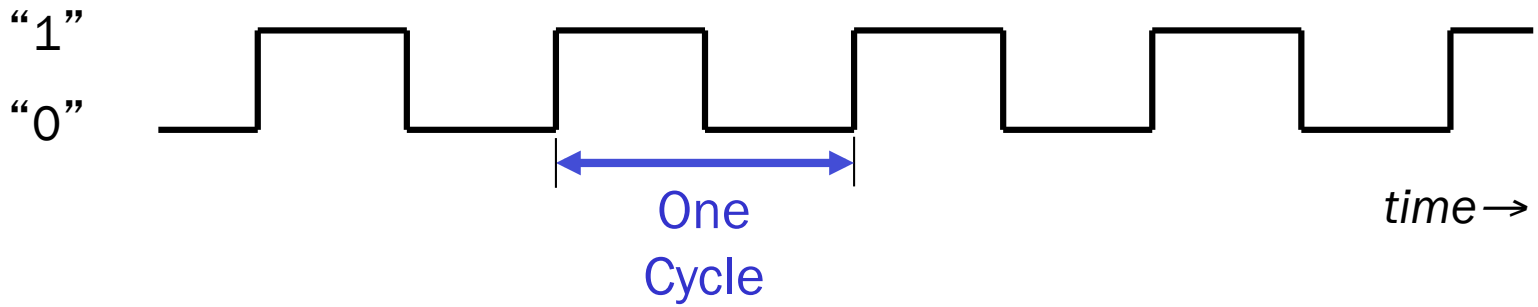
1. A finite number of states
2. A finite number of external inputs
3. A finite number of external outputs
4. An explicit specification of all state transitions
5. An explicit specification of what determines each external output value

Often described by a state diagram.

- Inputs trigger state transitions.
- Outputs are associated with each state (or with each transition).

# The Clock

Frequently, a clock circuit triggers transition from one state to the next.




At the beginning of each clock cycle,
state machine makes a transition,
based on the current state and the external inputs.

- **Not always required. In lock example, the input itself triggers a transition.**

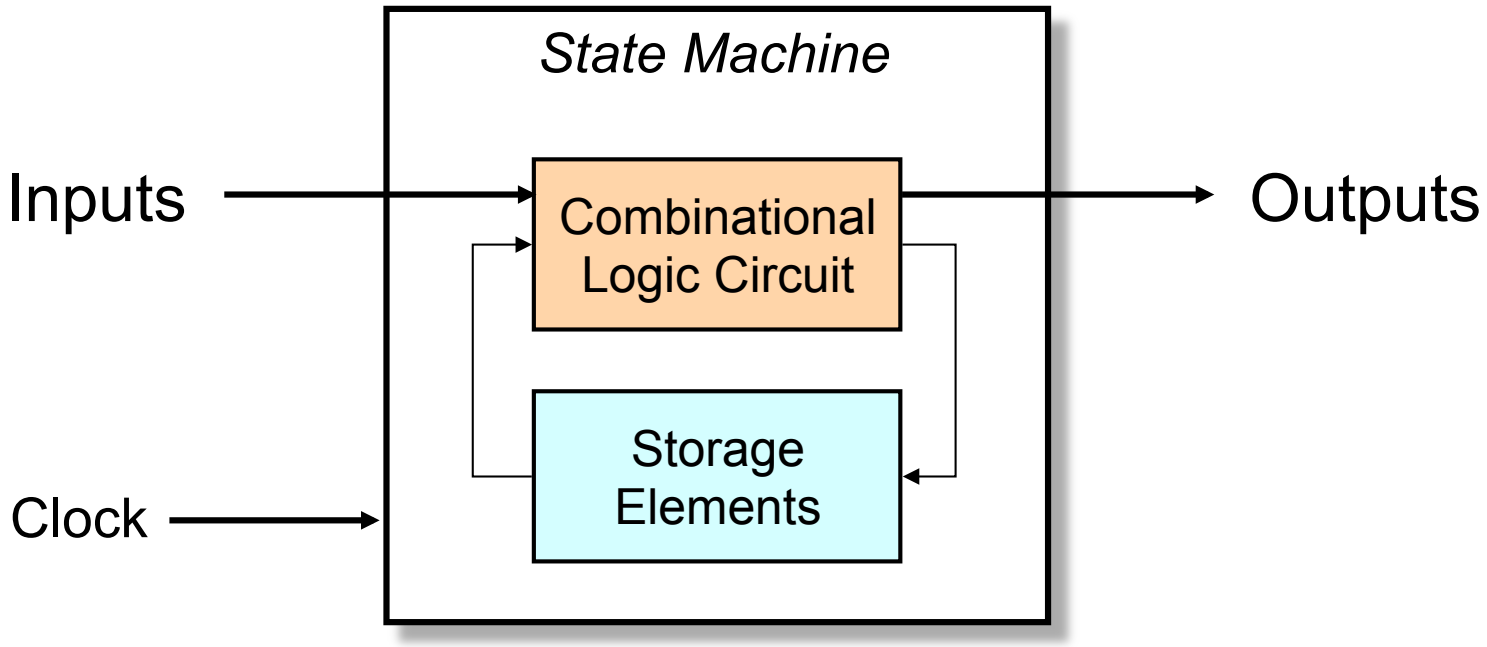
# Implementing a Finite State Machine

## Combinational logic

- Determine outputs and next state.

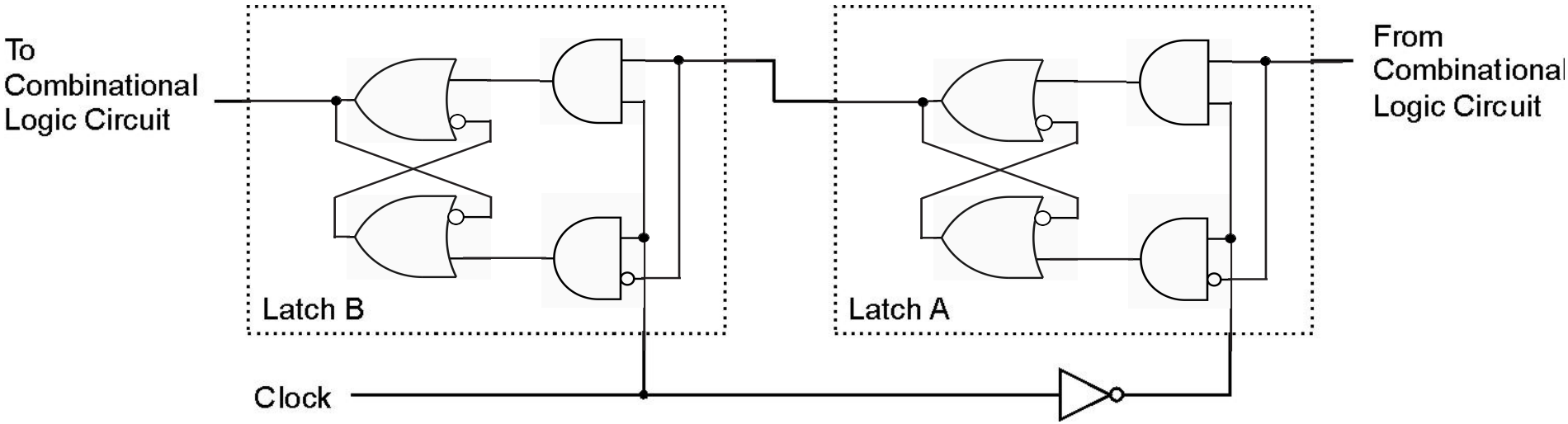
## Storage elements

- Maintain state representation.



# Storage: Master-Slave Flipflop

A pair of gated D-latches,
to isolate *next* state from *current* state.




During 1st phase (clock=1), previously-computed state becomes *current* state and is sent to the logic circuit.

During 2nd phase (clock=0), *next* state, computed by logic circuit, is stored in Latch A.

# Storage

Each master-slave flipflop stores one state bit.

The number of storage elements (flipflops) needed
is determined by the number of states
(and the representation of each state).

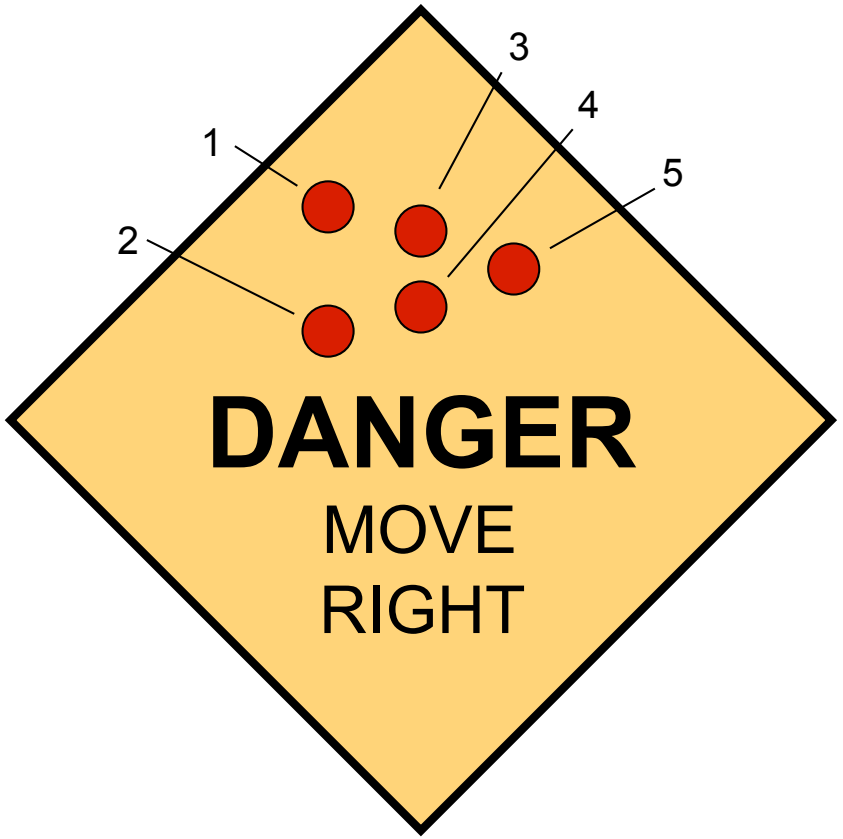
Examples:

- Sequential lock
  - Four states – two bits
- Basketball scoreboard
  - 7 bits for each score, 5 bits for minutes, 6 bits for seconds, 1 bit for possession arrow, 1 bit for half, …

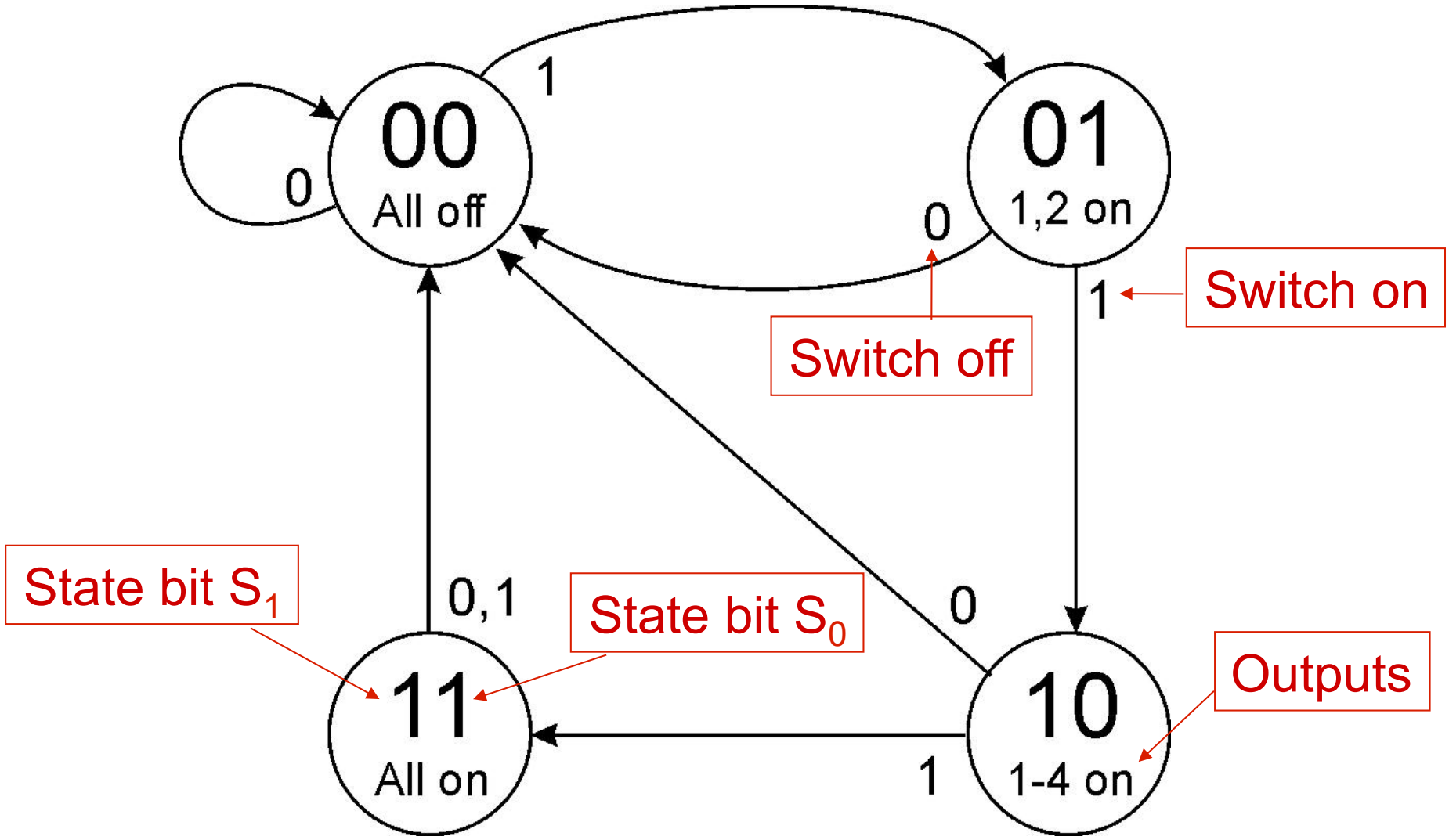
# Complete Example

A blinking traffic sign

- No lights on
- 1 & 2 on
- 1, 2, 3, & 4 on
- 1, 2, 3, 4, & 5 on
- (repeat as long as switch is turned on)



# Traffic Sign State Diagram




*Transition on each clock cycle*

# Traffic Sign Truth Tables

## Outputs (depend only on state: $S_1S_0$)

Lights 1 and 2
Lights 3 and 4
Light 5

| $S_1$ | $S_0$ | Z | Y | X |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

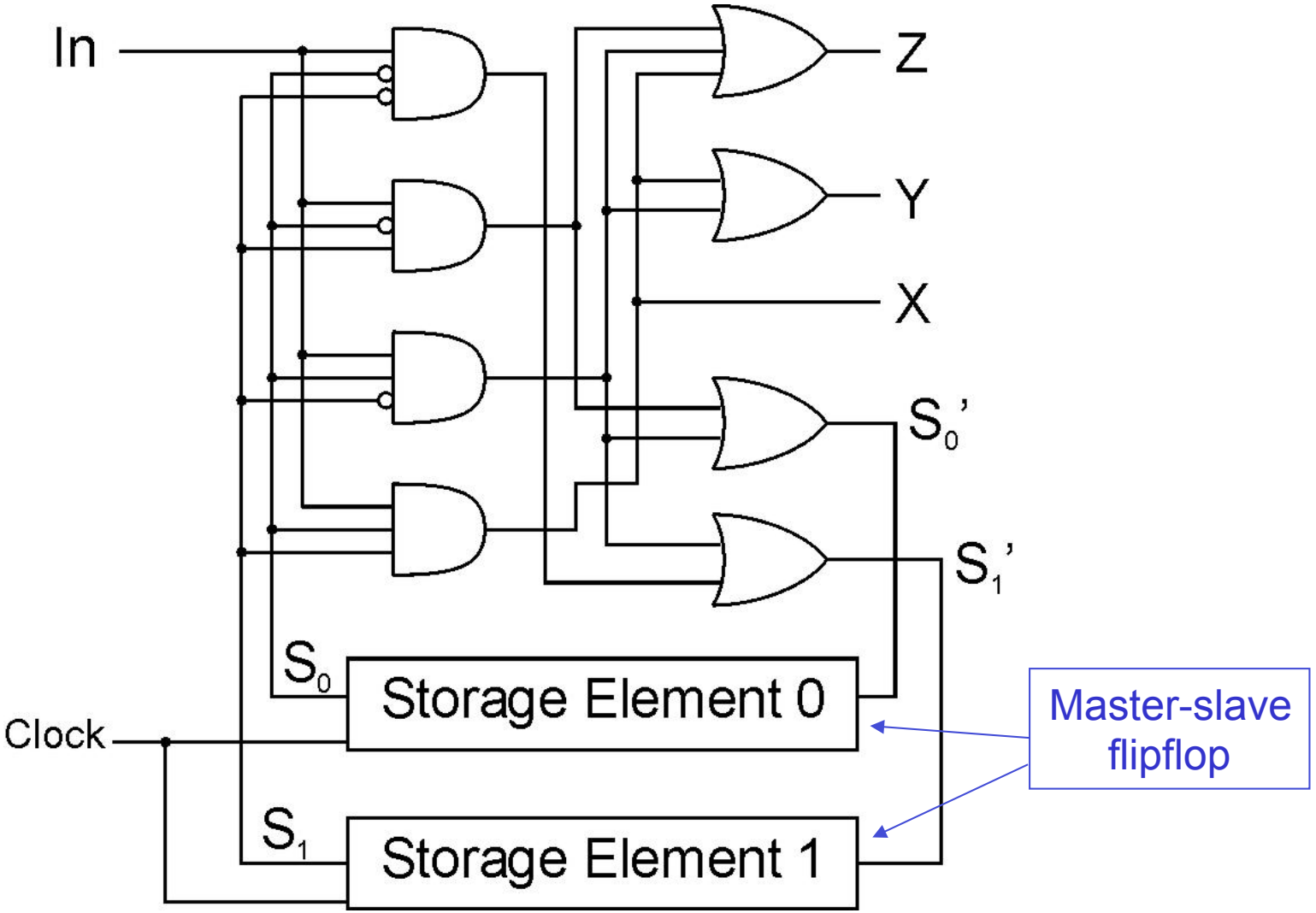
## Next State: $S_1'S_0'$ (depend on state and input)

Switch

| On | $S_1$ | $S_0$ | $S_1'$ | $S_0'$ |
|---|---|---|---|---|
| 0 | X | X | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |

Whenever On=0, next state is 00.

# Traffic Sign Logic



# From Logic to Data Path

The data path of a computer is all the logic used to process information.

- See the data path of the LC-3 on next slide.

## Combinational Logic

- Decoders – convert instructions into control signals
- Multiplexers – select inputs and outputs
- ALU (Arithmetic and Logic Unit) – operations on data

## Sequential Logic

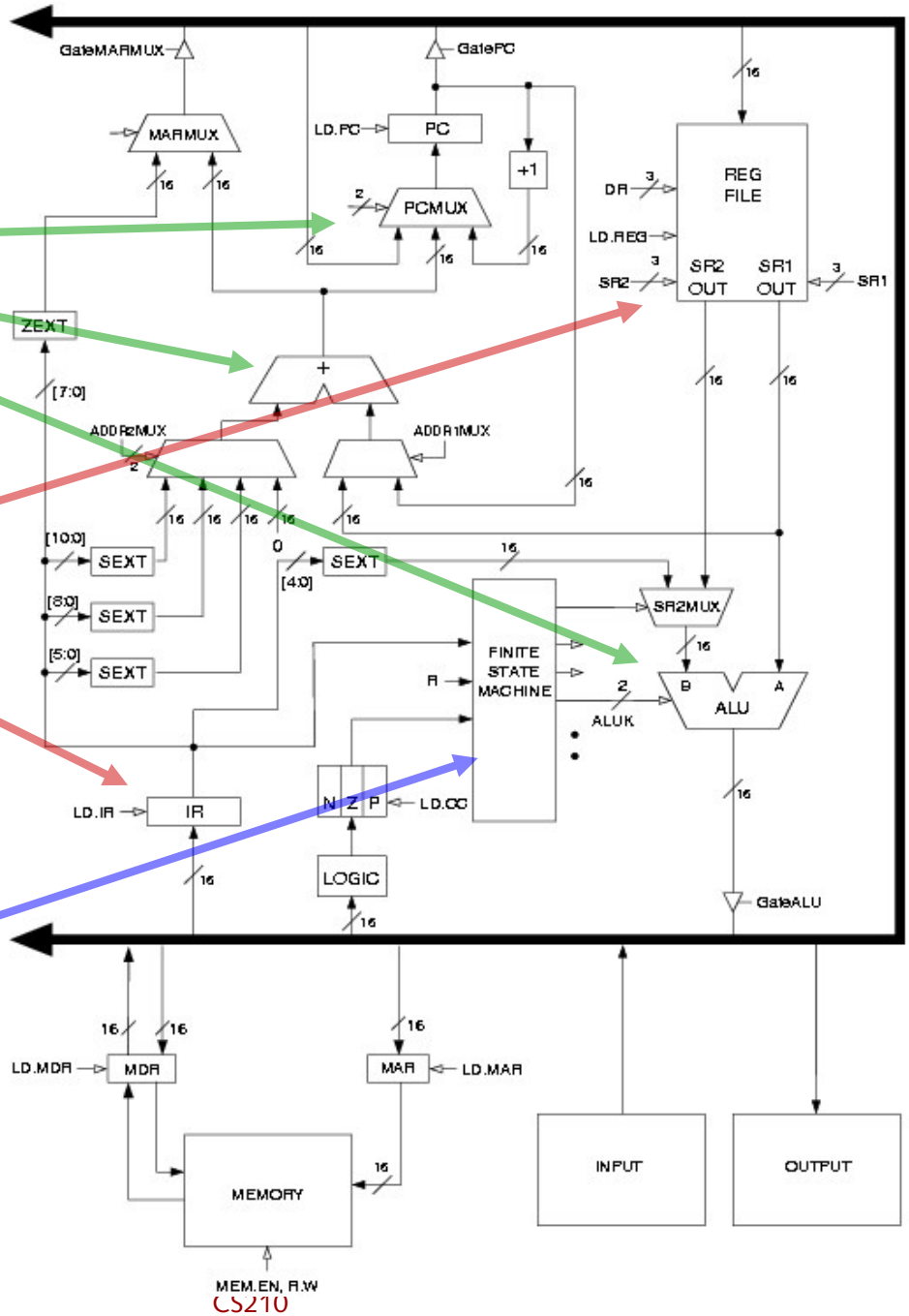
- State machine – coordinate control signals and data movement
- Registers and latches – storage elements

# LC-3 Data Path

Combinational Logic

Storage

State Machine



Computer Science 210 s1c

# Computer Systems 1

2013 Semester 1

Lecture Notes

## Chapter 4:

# The von Neumann Model

*James Goodman*


Credits: "McGraw-Hill" slides prepared by Gregory T. Byrd, North Carolina State University

# The Stored Program Computer

## 1943: Colossus Mark 1

- First programmable computer
- Specifically for breaking German codes.
- Secret until 1970s!

## 1943: ENIAC

- Hard-wired program – settings of dials and switches.
- Presper Eckert and John Mauchly – first general-purpose electronic computer.
  *(or was it John V. Atanasoff in 1942?)*
  *(or was it Konrad Zuse in 1941?)*

## 1944: Beginnings of EDSAC

- Maurice Wilkes, inspired by conversations with Eckert & Mauchly
- Among other improvements, includes program stored in memory

## 1944: Beginnings of EDVAC (working 1949)

For more history, see http://www.maxmon.com/history.htm

# The Stored Program Computer (continued)

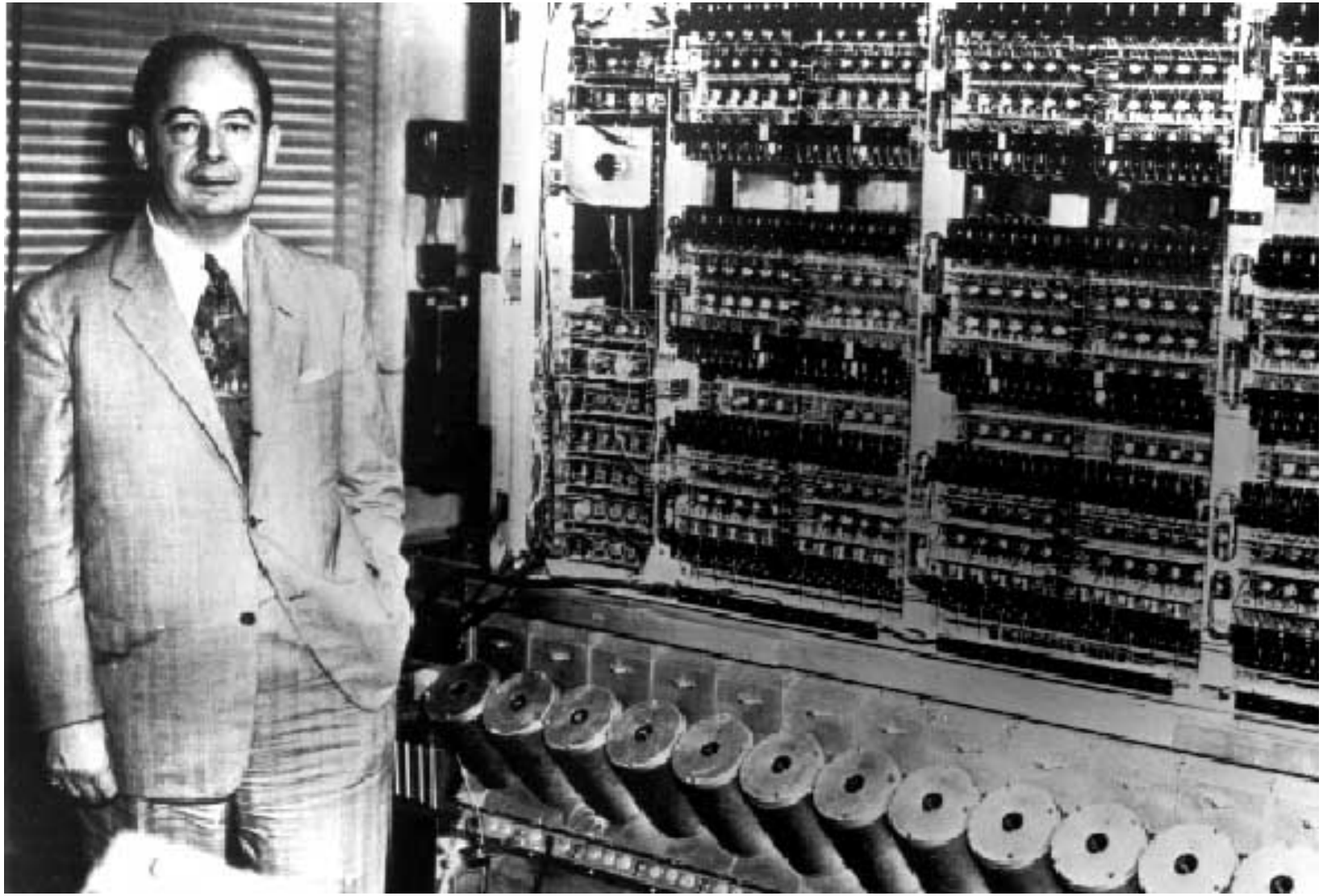
1945: John von Neumann

- Wrote a report on the stored program concept, known as the *First Draft of a Report on EDVAC*

The basic structure proposed in the draft became known as the "von Neumann machine" (or model).

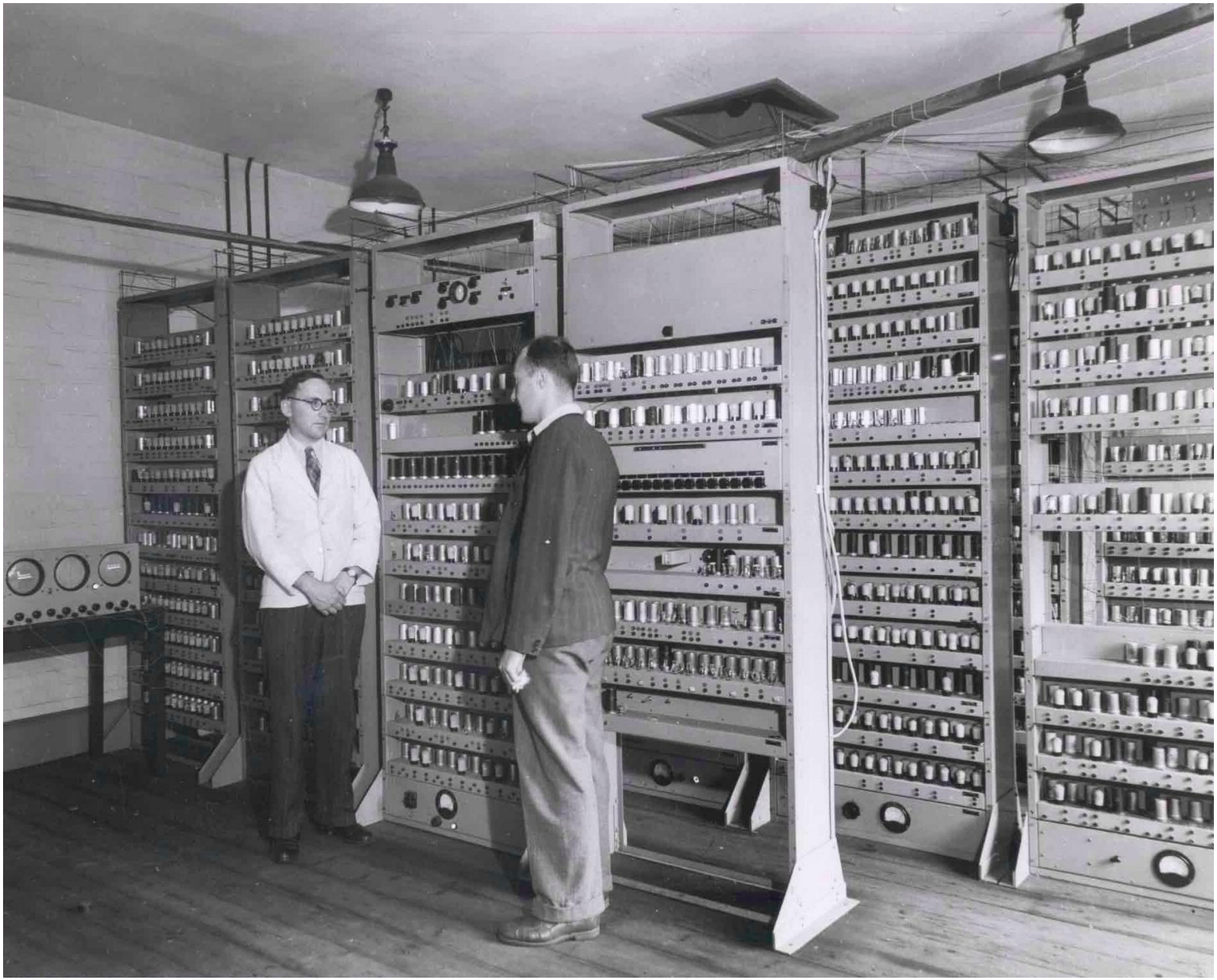
- a *memory*, containing instructions and data
- a *processing unit*, for performing arithmetic and logical operations
- a *control unit*, for interpreting instructions

For more history, see http://www.maxmon.com/history.htm



John von Neumann (1903-1957) with the ENIAC

# Konrad Zuse, 1910-1995



Maurice Wilkes (1913-2011), designer of the EDSAC, the first working stored-program computer (1949)

# Von Neumann Model