CPT_S 260 Intro to Computer Architecture
Lecture 30

Exam 2 Review
March 30, 2022

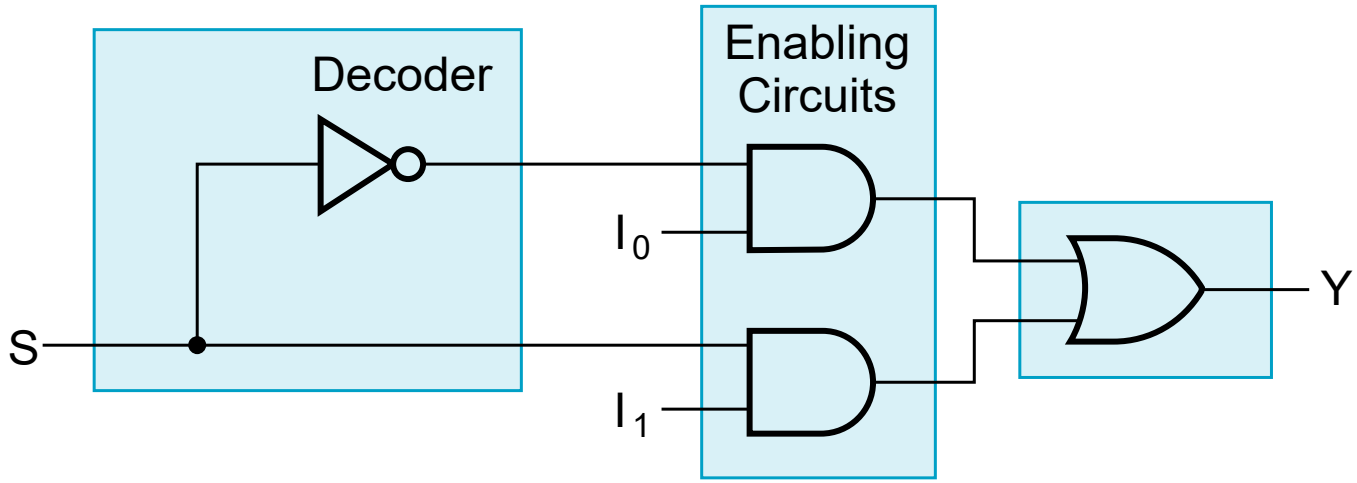
Ganapati Bhat
School of Electrical Engineering and Computer Science
Washington State University

2-to-1-Line Multiplexer

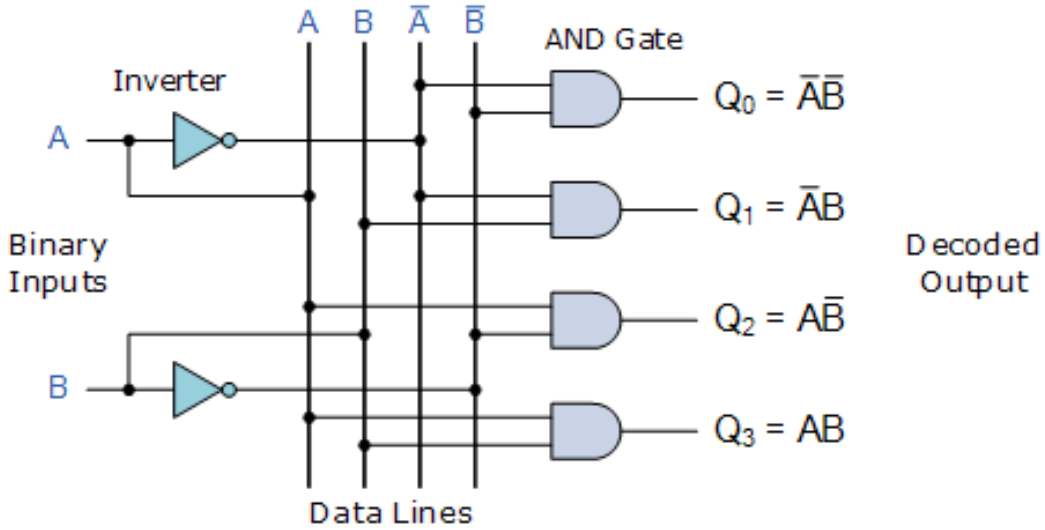
- **Since 2 = 2^1, n = 1**
- **The single selection variable S has two values:**
  - S = 0 selects input I_0
  - S = 1 selects input I_1
- **The equation:**

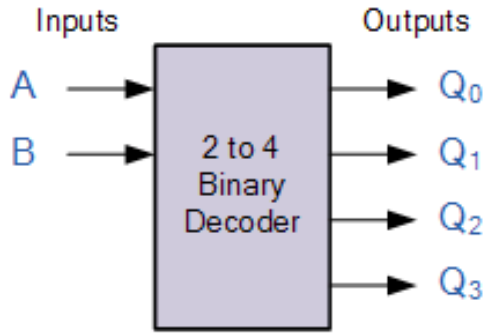
$$Y = \bar{S}I_0 + SI_1$$

- **The circuit:**



Decoder





Truth Table

A	B	Q_0	Q_1	Q_2	Q_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Rules of Boolean Algebra

- **Associative Law of multiplication**

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C$$

- **Distributive Law of multiplication**

$$A + BC = (A + B) \cdot (A + C)$$

- **Annulment law:**

$$A \cdot 0 = 0$$
$$A + 1 = 1$$

- **Identity law:**

$$A \cdot 1 = A$$
$$A + 0 = A$$

Rules of Boolean Algebra

- **Complement law:**

$$A + \bar{A} = 1$$
$$A \cdot \bar{A} = 0$$

- **Double negation law:**

$$\bar{\bar{A}} = A$$

- **Absorption law:**

$$A \cdot (A + B) = A$$
$$A + AB = A$$
$$A + \bar{A}B = A + B$$

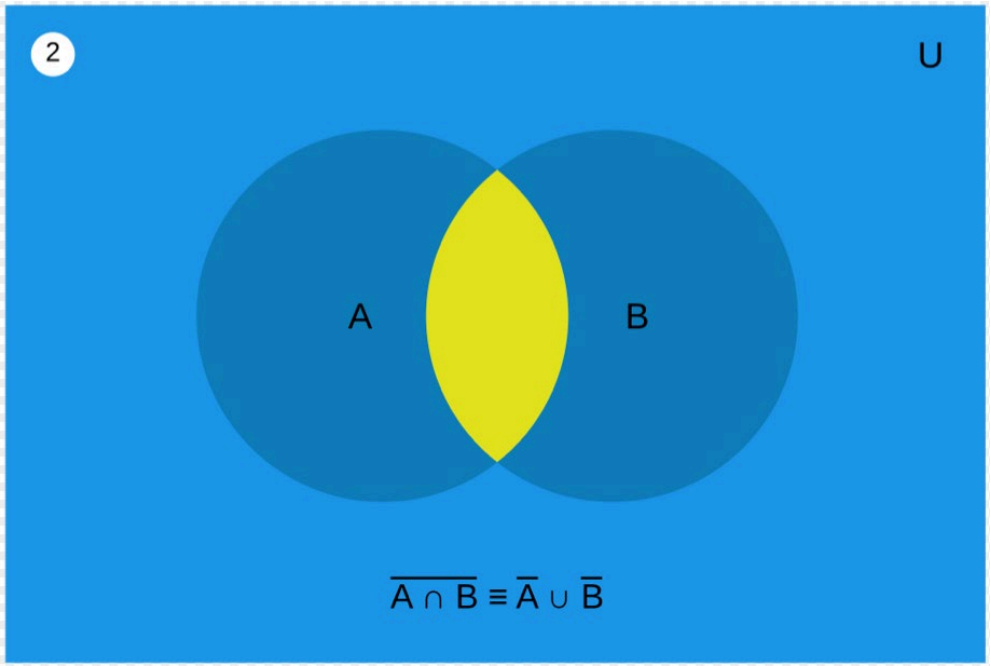
- **Idempotent law:**

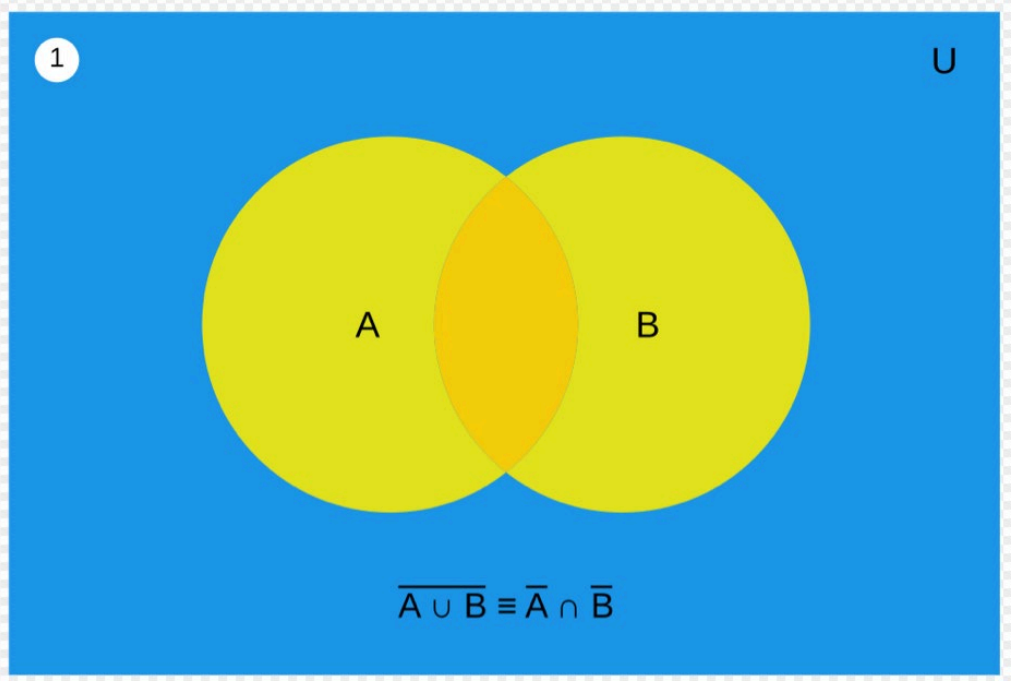
$$A + A = A$$
$$A \cdot A = A$$

De Morgan's Laws

- **Transformation rules that help simplification of negations**
- **Statement:**

$$\overline{AB} = \bar{A} + \bar{B}$$
$$\overline{(A + B)} = \bar{A} \cdot \bar{B}$$





Sum of Products

- **Minterm Expressions**
- **If input is 0 we take the complement of the variable**
- **If input is 1 we take the variable as is**
- **To get the desired canonical SOP expression we will add the minterms (product terms) for which the output is 1**

$$F = \bar{A}B + A\bar{B} + AB$$

A	B	F	Minterm
0	0	0	A'B'
0	1	1	A'B
1	0	1	AB'
1	1	1	AB

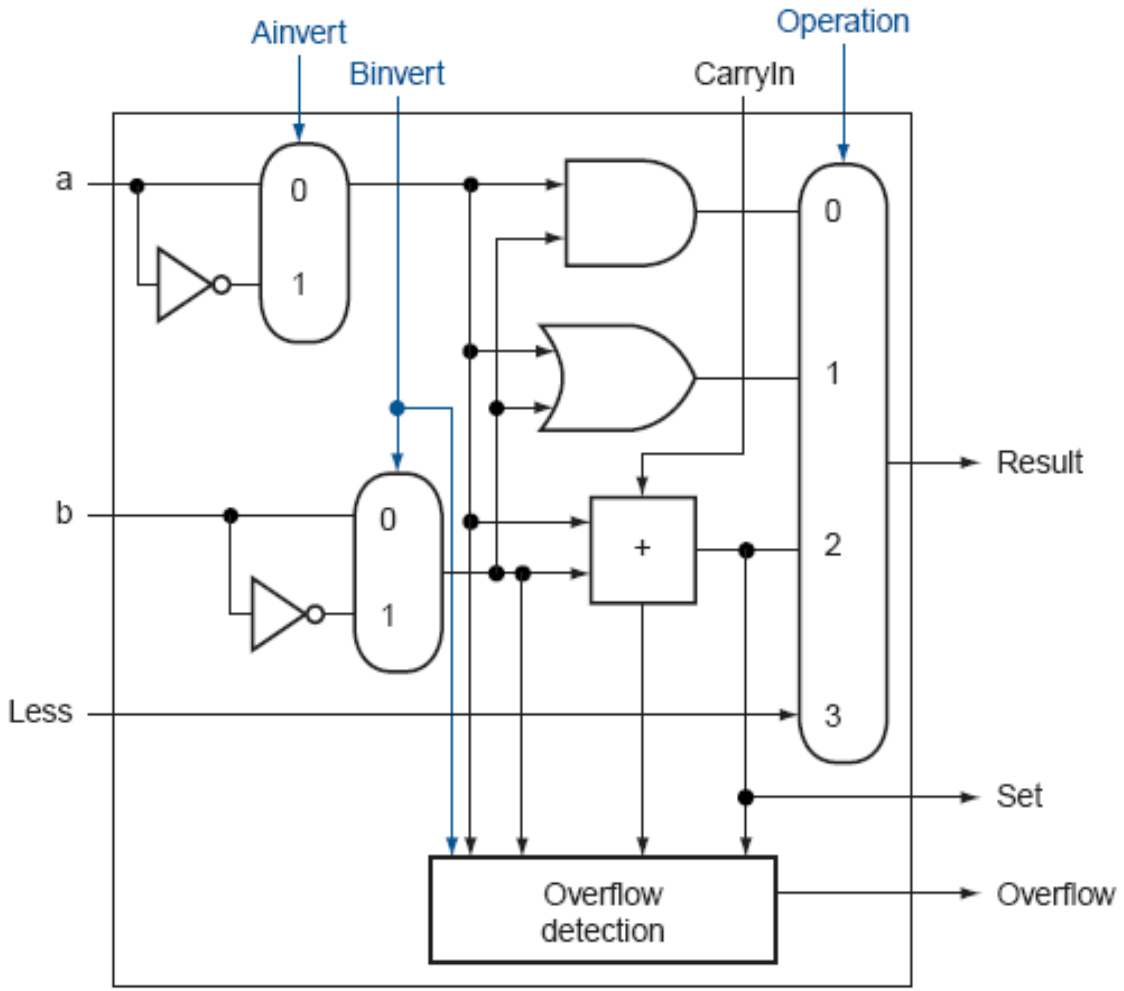
Product of Sums

- **Maxterm Expressions**
- **If input is 1, we take the complement of the variable**
- **If input is 0, we take the variable as is**
- **To get the desired canonical POS expression we will multiply the maxterms (sum terms) for which the output is 0**

$$F = (A + B) . (\bar{A} + \bar{B})$$

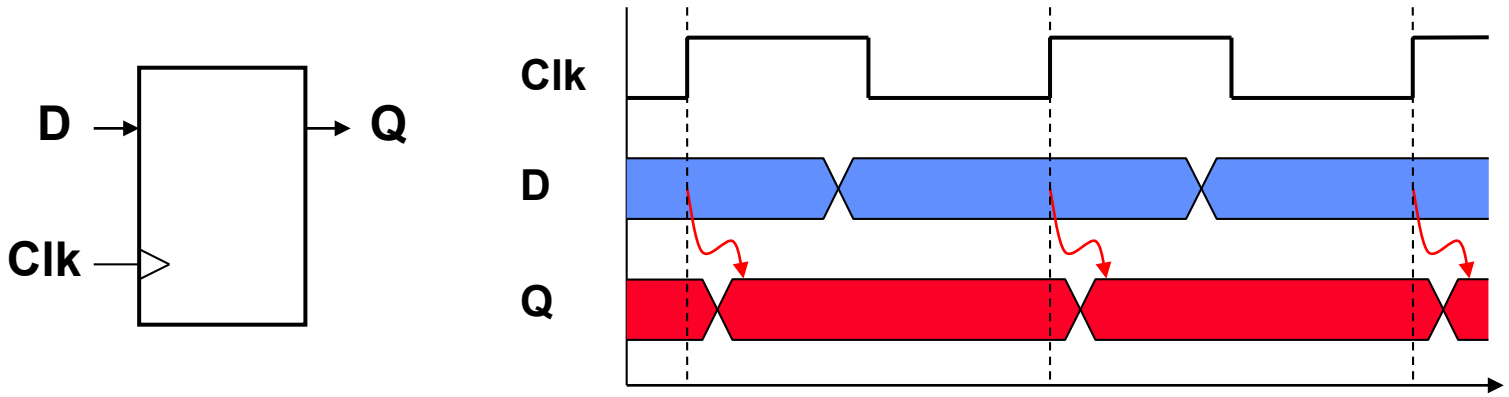
A	B	F	Minterm
0	0	0	A'B'
0	1	1	A'B
1	0	1	AB'
1	1	1	AB

1-bit ALU


Ainvert
Binvert
CarryIn
Operation
a
b
Less
0
1
0
1
0
1
2
3
+
Result
Set
Overflow
Overflow detection

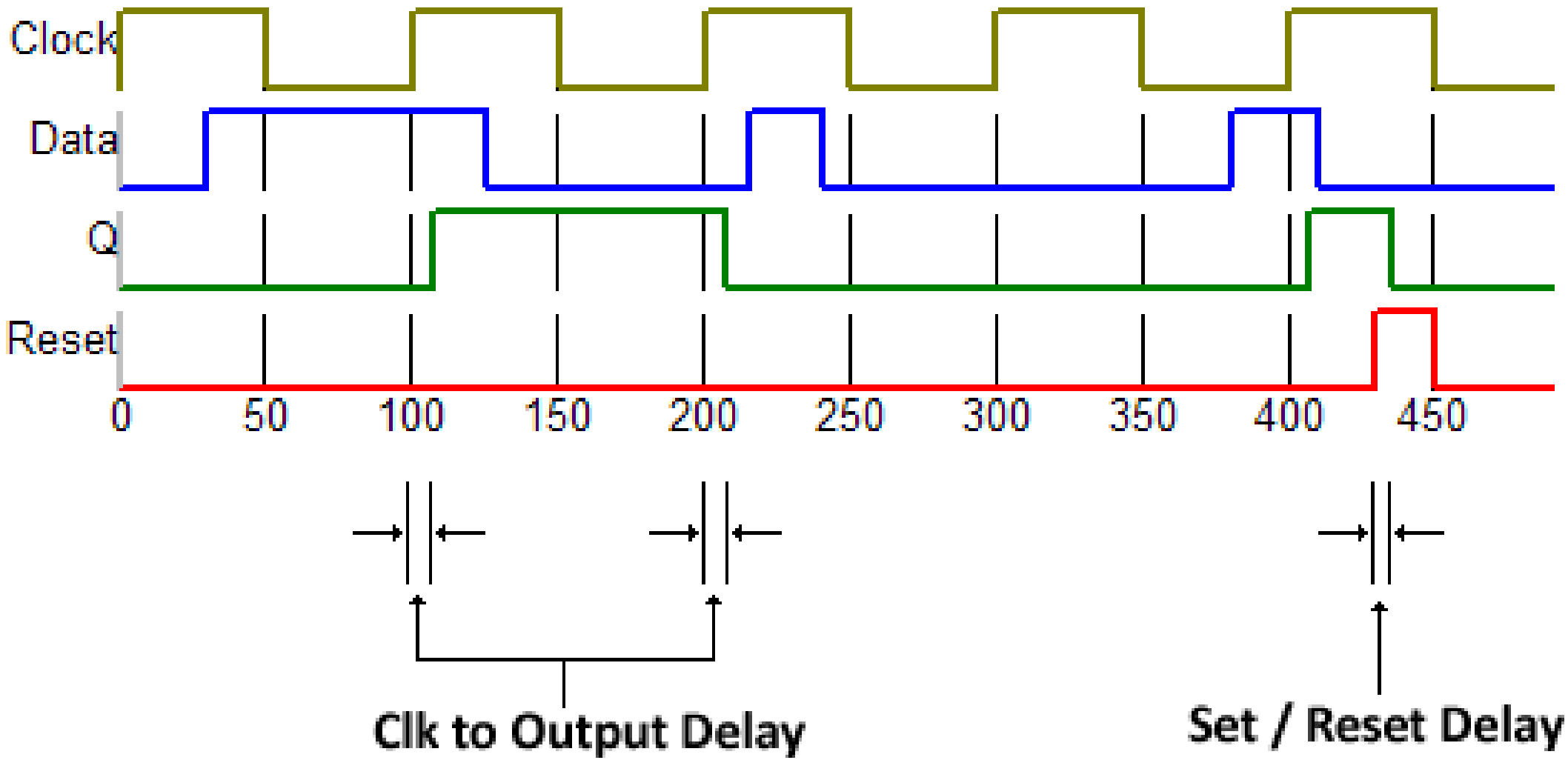
Sequential Elements

- **Flip flops: stores data in a circuit**
  - Uses a clock signal to determine when to update the stored value
  - Edge-triggered: update when Clk changes from 0 to 1



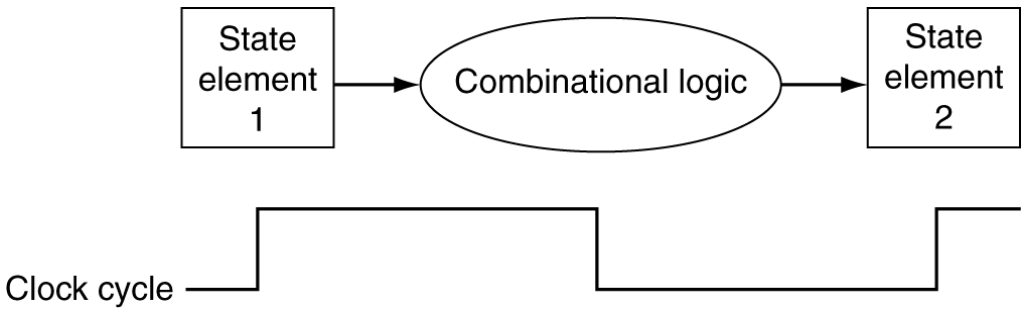
10

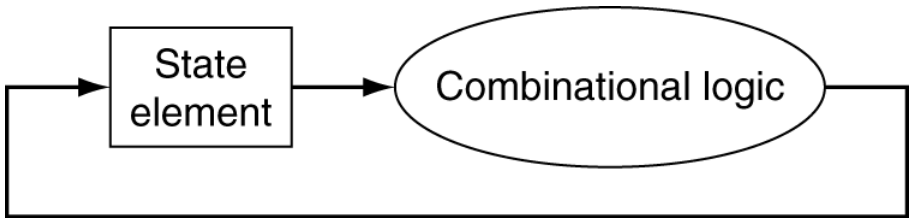
Clocking Sequence



Clocking Methodology

- **Combinational logic transforms data during clock cycles**
  - Between clock edges
  - Input from state elements, output to state element
  - Longest delay determines clock period





MIPS Datapath