

CS126 Precept 6

Meeting 10: sequential circuits,
TOY architecture, and exam

Randy Wang

- Exam archive on-line
- Q and A review session Tuesday March 23, 7:30pm, CS105
- Exam Wednesday, March 24, 7:30pm, Friend 101
- Closed book, closed notes except for one 8.5 x 11 single-sided handwritten cheat sheet.
- Topics list: anything from lecture, precept, homework and reading considered fair game

- * Java programming basics - only Fall 2003 has java questions, but previous C test questions have similar topics and difficulty.
 - Conditionals and Loops
 - Functions
 - Arrays
 - Recursion
 - Debugging

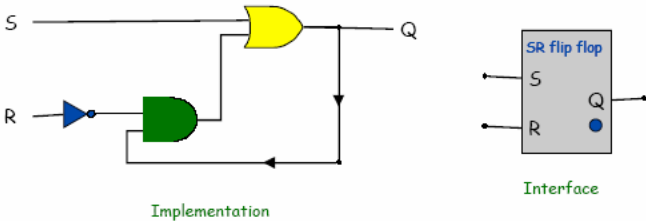
- * Number Systems - decimal, binary, hex
- * TOY Programming - TOY cheat sheet will be provided
 - Current version of TOY used since Fall 2001.
- * Boolean logic represented by algebraic expressions, truth tables, logic gates (e.g., AND, OR, NOT, XOR)
- * Combinational and Sequential Circuits
 - Function of decoder, mux, adder
 - Function of flip-flop, register, memory, counter
 - Interpreting circuits via truth table and/or timing diagram
- * Machine Architecture - functional blocks, data flow

Review SR Flip-Flop

SR Flip-Flop

SR Flip-Flop.

- $S = 1, R = 0$ (set) \Rightarrow Flips "bit" on.
- $S = 0, R = 1$ (reset) \Rightarrow Flips "bit" off.
- $S = R = 0$ \Rightarrow Status quo.
- $S = R = 1$ \Rightarrow Not allowed.



Review SR Flip-Flop

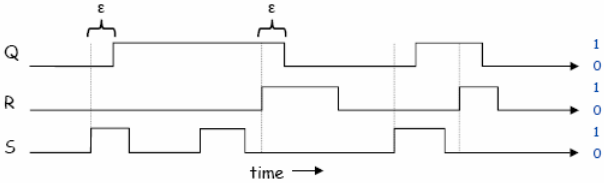
Truth Table and Timing Diagram

Truth table.

- Values vary over time.
- $S(t), R(t), Q(t)$ denote value at time t .

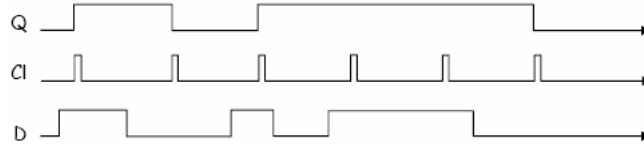
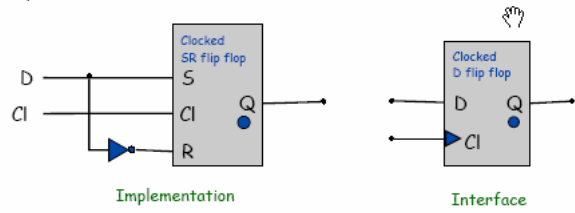
SR Flip Flop Truth Table			
S(t)	R(t)	Q(t)	Q(t+ ϵ)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	
1	1	1	

Sample timing diagram for SR flip-flop.

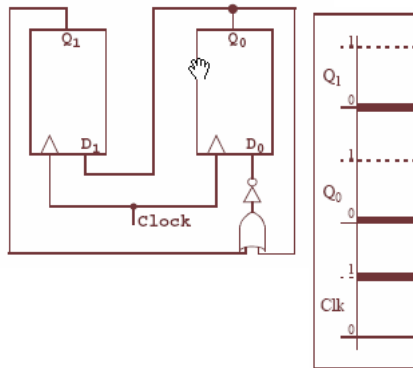


Review Clocked D Flip-Flop

- Output follows D input while clock is 1.
- Output is remembered while clock is 0.



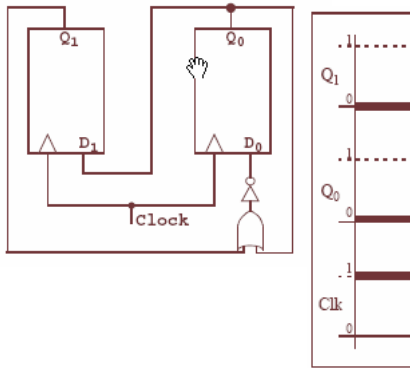
Tracing Timing Diagram



$$Q_0(t_0) = 0$$

$$Q_1(t_0) = 0$$

Tracing Timing Diagram



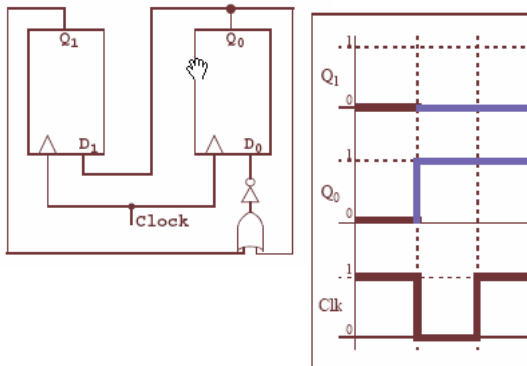
$$Q_0(t_0) = 0$$

$$Q_1(t_0) = 0$$

$$Q_0(t_1) = (Q_0 + Q_1)' = (0 + 0)' = 1$$

$$Q_1(t_1) = Q_0 = 0$$

Tracing Timing Diagram



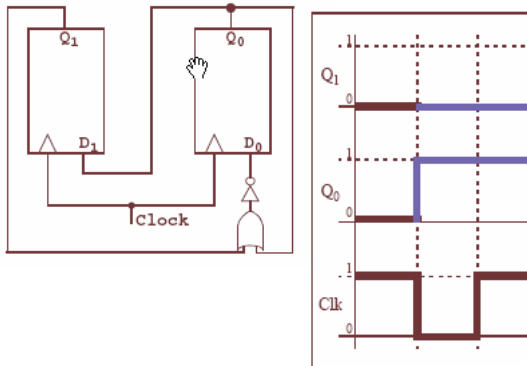
$$Q_0(t_0) = 0$$

$$Q_1(t_0) = 0$$

$$Q_0(t_1) = (Q_0 + Q_1)' = (0 + 0)' = 1$$

$$Q_1(t_1) = Q_0 = 0$$

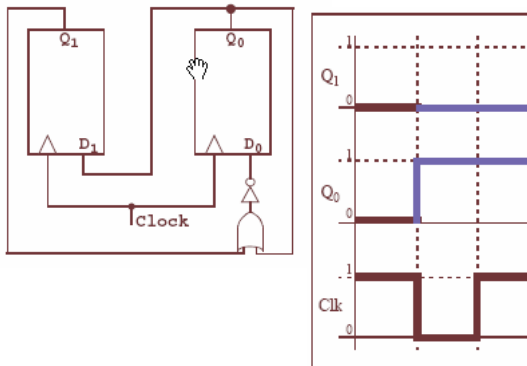
Tracing Timing Diagram



$$Q_0(t_1) = 1$$

$$Q_1(t_1) = 0$$

Tracing Timing Diagram



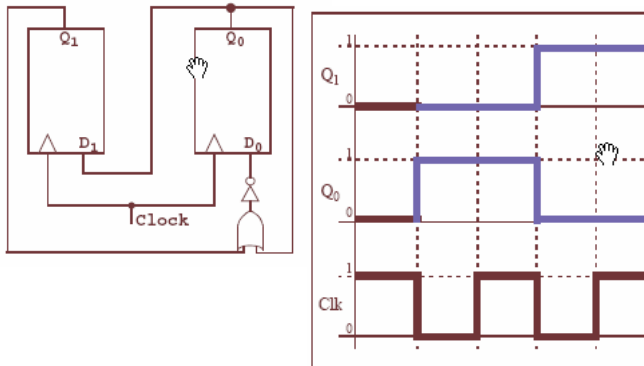
$$Q_0(t_1) = 1$$

$$Q_1(t_1) = 0$$

$$Q_0(t_2) = (Q_0 + Q_1)' = (1 + 0)' = 0$$

$$Q_1(t_2) = Q_0 = 1$$

Tracing Timing Diagram



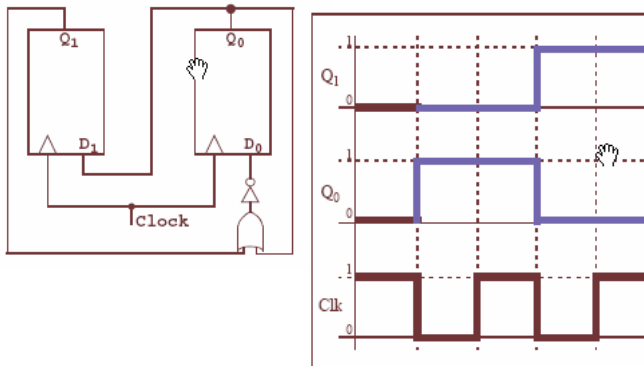
$$Q_0(t_1) = 1$$

$$Q_1(t_1) = 0$$

$$Q_0(t_2) = (Q_0 + Q_1)' = (1 + 0)' = 0$$

$$Q_1(t_2) = Q_0 = 1$$

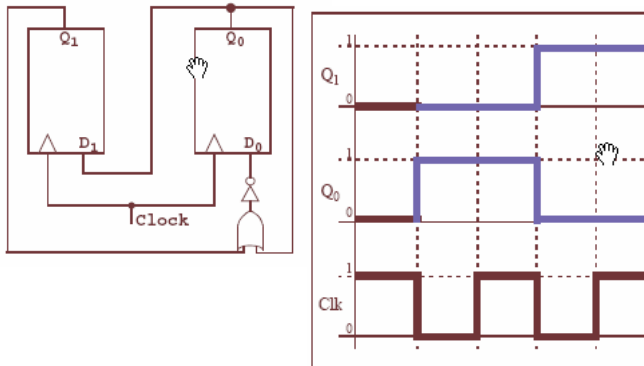
Tracing Timing Diagram



$$Q_0(t_2) = 0$$

$$Q_1(t_2) = 1$$

Tracing Timing Diagram



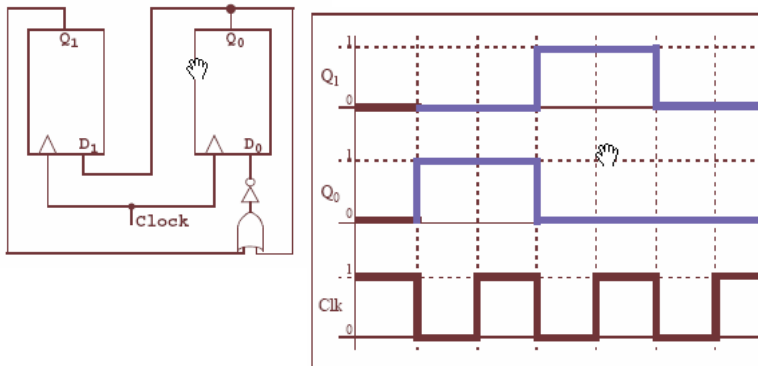
$$Q_0(t_2) = 0$$

$$Q_1(t_2) = 1$$

$$Q_0(t_3) = (Q_0 + Q_1)' = (0 + 1)' = 0$$

$$Q_1(t_3) = Q_0 = 0$$

Tracing Timing Diagram



$$Q_0(t_2) = 0$$

$$Q_1(t_2) = 1$$

$$Q_0(t_3) = (Q_0 + Q_1)' = (0 + 1)' = 0$$

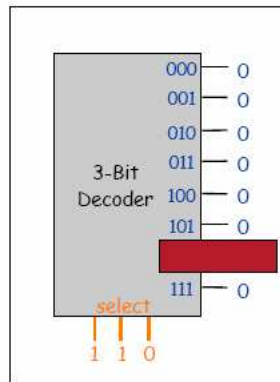
$$Q_1(t_3) = Q_0 = 0$$

n-Bit Decoder

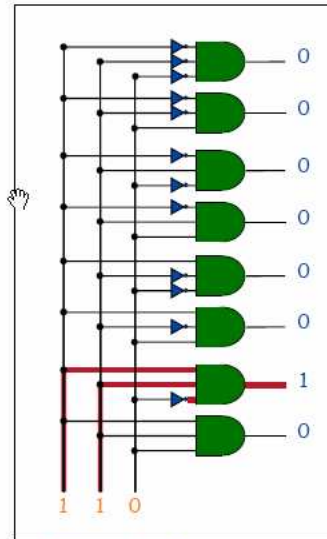
n = 8 for main memory

n-bit decoder.

- n address inputs, 2^n data outputs.
- Addressed output bit is 1; others are 0.



3-Bit Decoder Interface



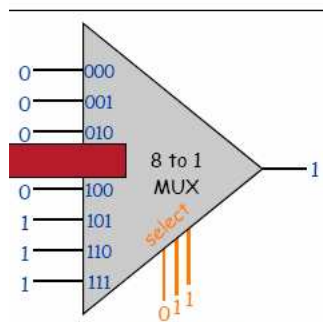
3-Bit Decoder Implementation

2^n -to-1 Multiplexer

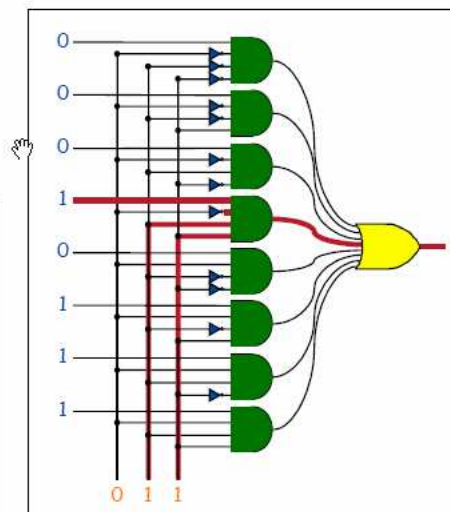
n = 8 for main memory

2^n -to-1 multiplexer.

- n select inputs, 2^n data inputs, 1 output.
- Copies "selected" data input bit to output.



8-to-1 Mux Interface



8-to-1 Mux Implementation

Register File Implementation

Implementation example: TOY main memory.

- Use 256 16-bit registers.
- Multiplexer and decoder are combinational circuits.

