0 Admin
1 Recap
2 ROM based FSM
3 Serial Data Transmission – synchronous/asynchronous

CS150 Fall 1999 Lecture 13 Copyright © 1999 The Regents of the University of California R. Fearing

1 RECAP

- Reflections/Termination

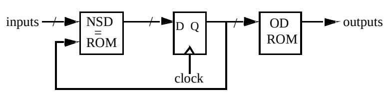
$$\Gamma = \frac{Z - Z_0}{Z + Z_0}$$

- ROM – LUT fixed at compile time → constant

- RAM – LUT programmable
  - set-up and hold times/tri-state
    level sensitive device, not edge triggered

CS150 Fall 1999 Lecture 13 Copyright © 1999 The Regents of the University of California R. Fearing

2 ROM-BASED FSM



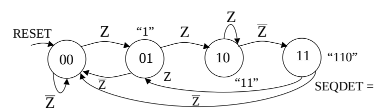
Advantage: don't need to fool with as much discrete logic

Disadvantage: can get carried away with huge number of states
not self-documenting, hard to modify incrementally

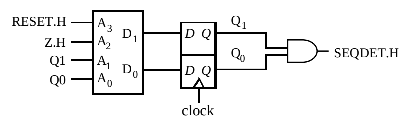
CS150 Fall 1999 Lecture 13 Copyright © 1999 The Regents of the University of California R. Fearing

2 ROM-BASED FSM (cont.)

1. Example ROM-based FSM



2. Schematic – Block diagram



CS150 Fall 1999 Lecture 13 Copyright © 1999 The Regents of the University of California R. Fearing

2 ROM-BASED FSM (cont.)

3. State table

Reset A3	Z A2	Q_1 Z1	Q_0 A0	Next State Q_1=D_1	Q_0=D_0	Output
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	1	0
0	0	1	1	0	0	1
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	1	0	0
0	1	1	1	0	1	1
1	X	X	X	0	0	$Q_1 \cdot Q_0$

CS150 Fall 1999 Lecture 13 Copyright © 1999 The Regents of the University of California R. Fearing

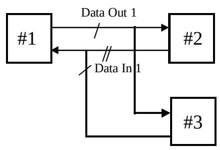
2 ROM-BASED FSM (cont.)

Address	Data	Address	Data
0	0	8	0
1	0	9	0
2	3	A	0
3	0	B	0
4	1	C	0
5	2	D	0
6	2	E	0
7	1	F	0

$\left.\right\}$ reset (addresses 8–F)

Note: Xilinx ROM – Values written from address 15 to 0,
not 0 → 15

3 SERIAL DATA TRANSMISSION

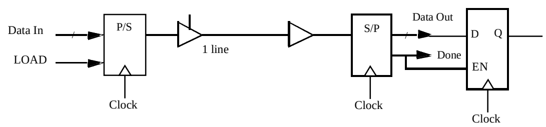


office building, or factory – distances ~1000 m

too many wires – need coax for each signal

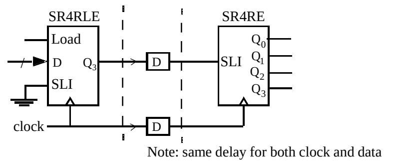
Alternative – Time division MUX → much fewer wires

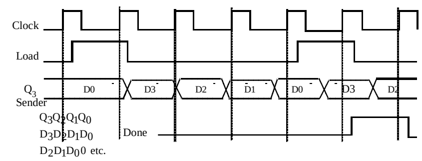
Basic idea



3 SERIAL DATA TRANSMISSION (cont.)

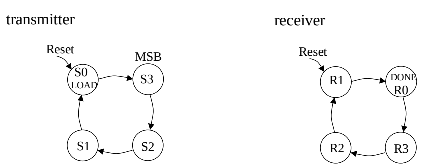
3.1 Synchronous Serial Data





3 SERIAL DATA TRANSMISSION (cont.)

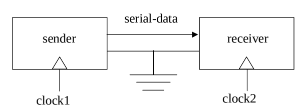
Need to synchronize transmitter and receiver



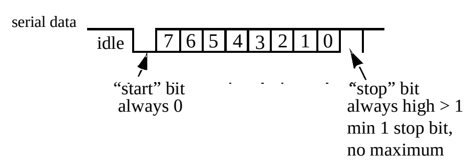
Problem: need extra wires to synchronize sender and receiver FSM

3 SERIAL DATA TRANSMISSION (cont.)

3.2 Asynchronous Serial Data Transmission



Idea: Add extra bits for synchronization



3 SERIAL DATA TRANSMISSION (cont.)

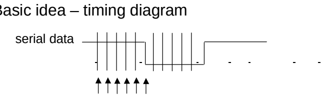
Serial transmitter

like synchronous, load all start and stop bits, e.g., use 10-bit loadable shift register

Receiver

1. needs to wait for start bit
2. needs to sample in middle of bit since timing may be off
3. input data is asynchronous – better synchronize
4. error detection – missing step
   – false start

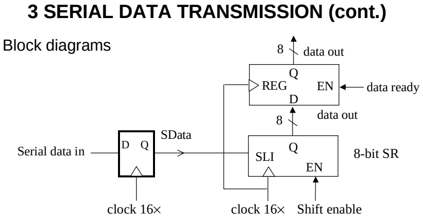
3.2.1 Basic idea – timing diagram



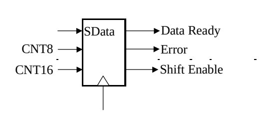
sample at high frequency to find start edge, wait half clock ...

3 SERIAL DATA TRANSMISSION (cont.)

3.2.2 Block diagrams



Control FSM



3 SERIAL DATA TRANSMISSION (cont.)

3.2.3 Initial State Diagram