CPE201
Digital Design

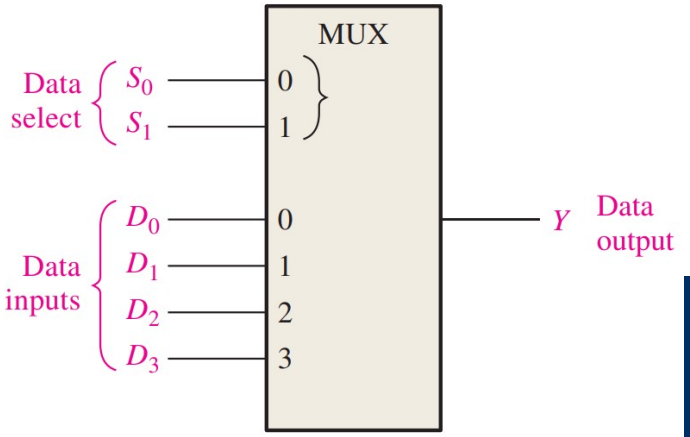
By Benjamin Haas

Class 16: Multiplexers and Parity

University of Nevada, Reno

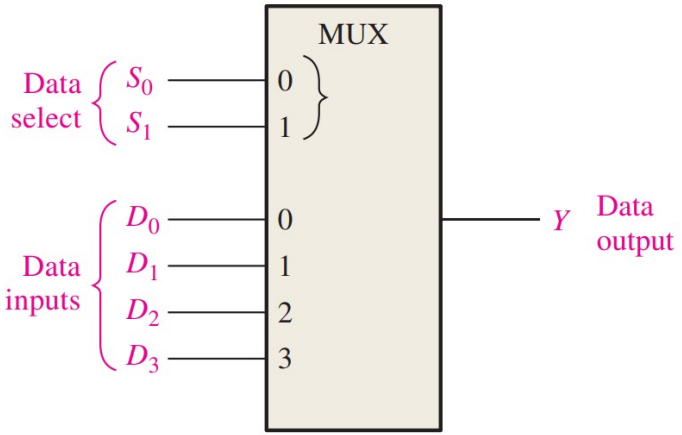
Multiplexer

- Usually called a MUX
- Has both encoder and decoder inputs
  - n selection inputs and 2^n data inputs
  - One output
  - n selects which of 2^n
    is on the output



MUX

Data-Select Inputs		Input Selected
S_1	S_0	
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3



MUX

The data output is equal to D_0 only if $S_1 = 0$ and $S_0 = 0$: $Y = D_0\overline{S}_1\overline{S}_0$.

The data output is equal to D_1 only if $S_1 = 0$ and $S_0 = 1$: $Y = D_1\overline{S}_1S_0$.

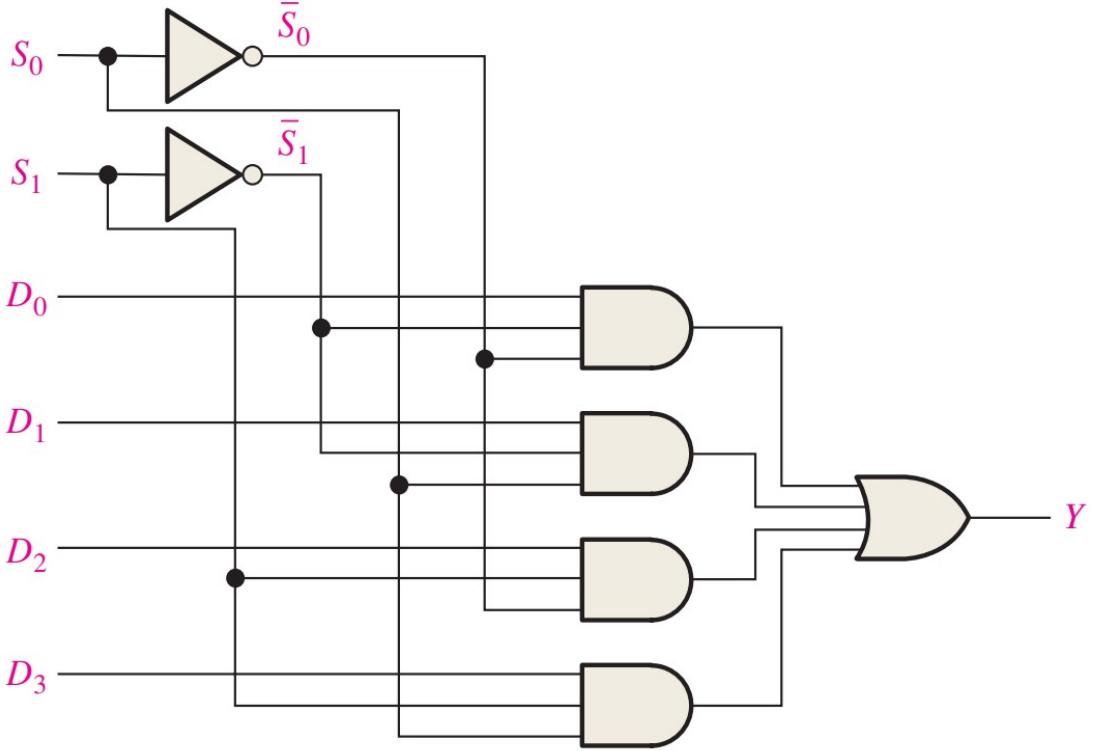
The data output is equal to D_2 only if $S_1 = 1$ and $S_0 = 0$: $Y = D_2S_1\overline{S}_0$.

The data output is equal to D_3 only if $S_1 = 1$ and $S_0 = 1$: $Y = D_3S_1S_0$.

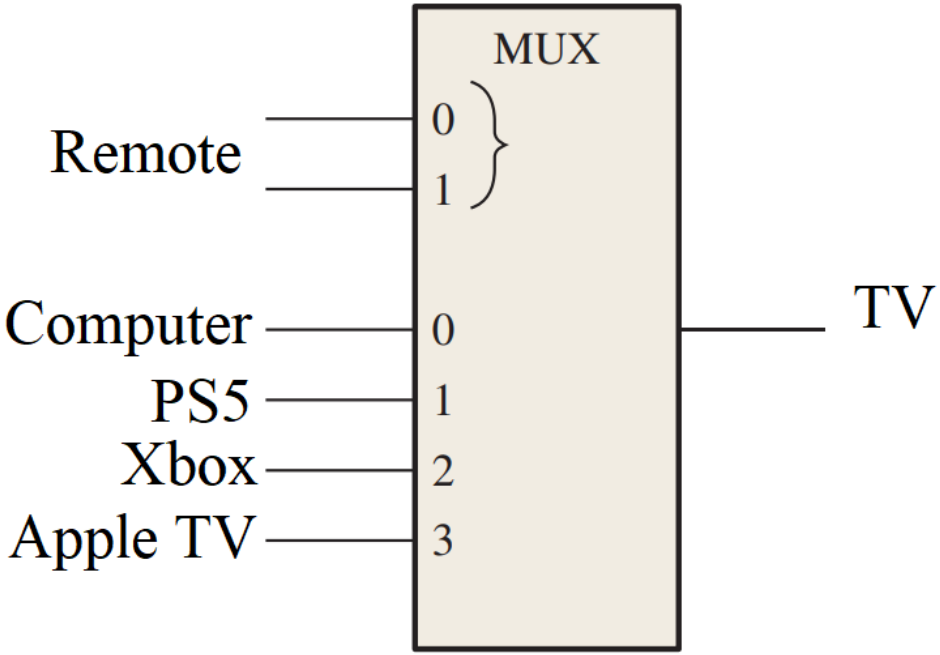
$$Y = D_0\overline{S}_1\overline{S}_0 + D_1\overline{S}_1S_0 + D_2S_1\overline{S}_0 + D_3S_1S_0$$

MUX

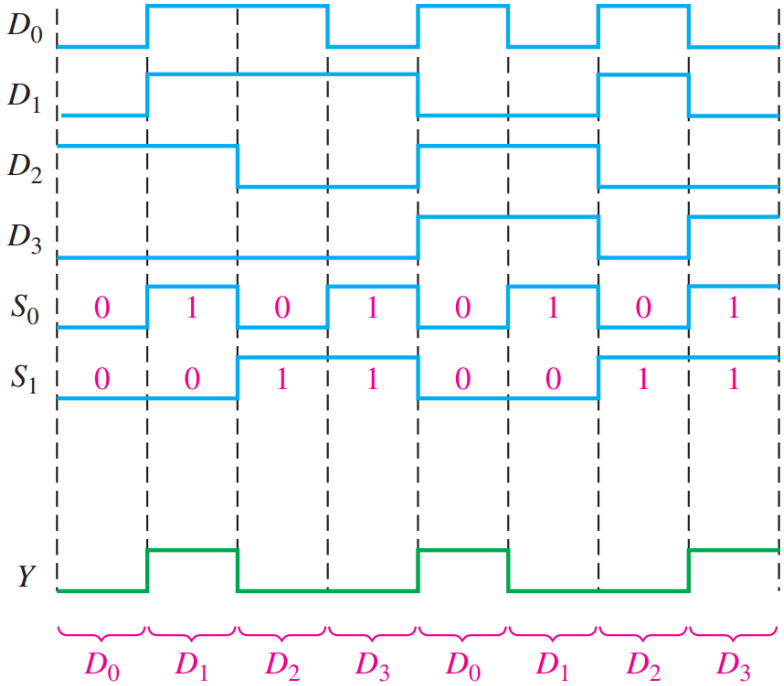
Data-Select Inputs		Input Selected
S_1	S_0	
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3



Example



Example

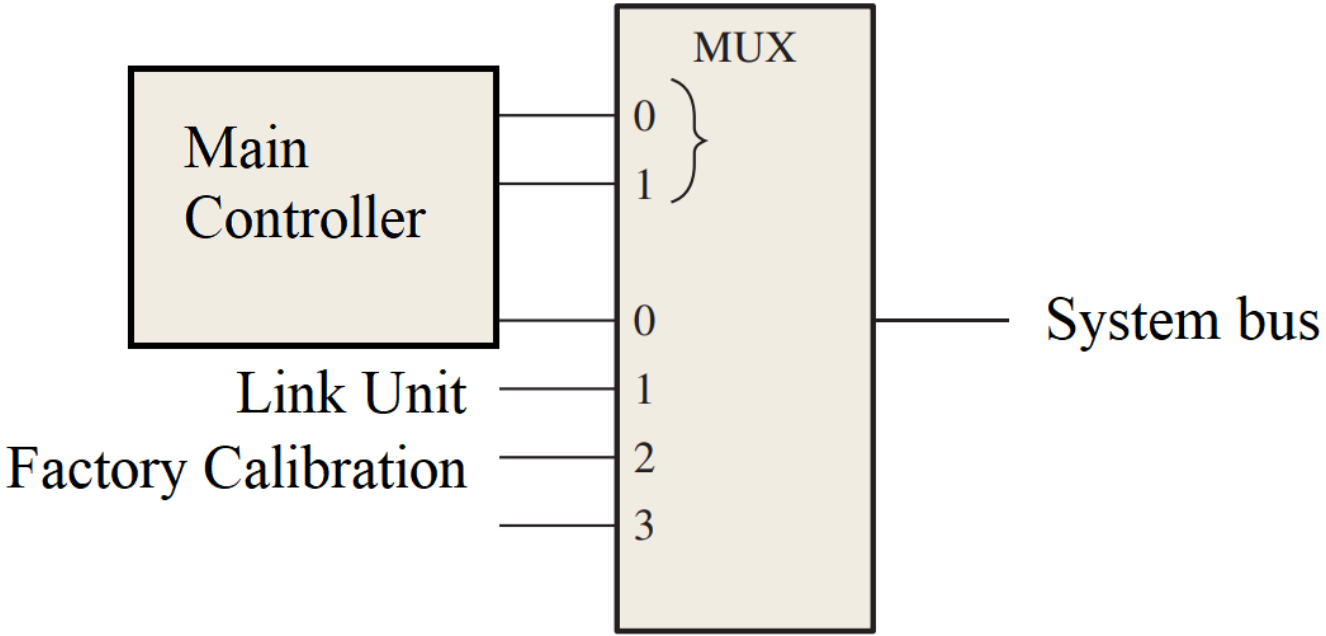


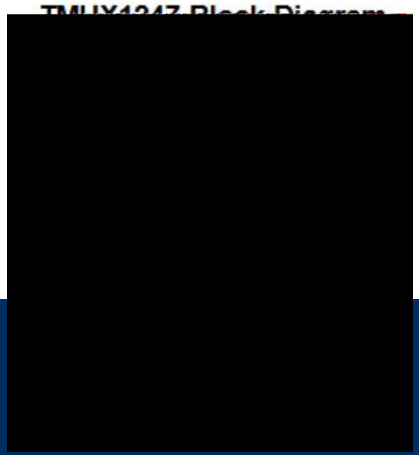
Transmission Modes

- Simplex Transmission Mode
  - Television, radio, above MUX circuit
- Half Duplex Transmission Mode
  - Walkie-talkie
- Full Duplex Transmission Mode
  - Phones

Half Duplex Example

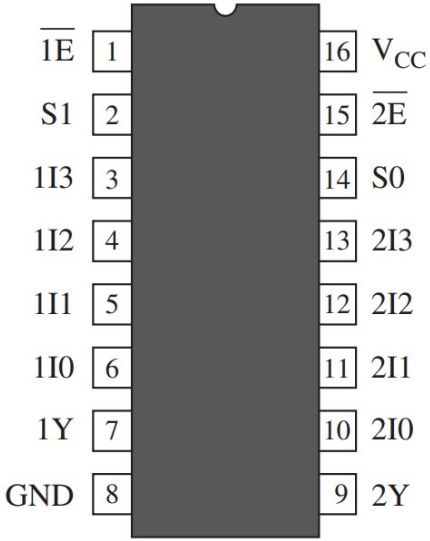
- Not really a MUX, but a MUX switch



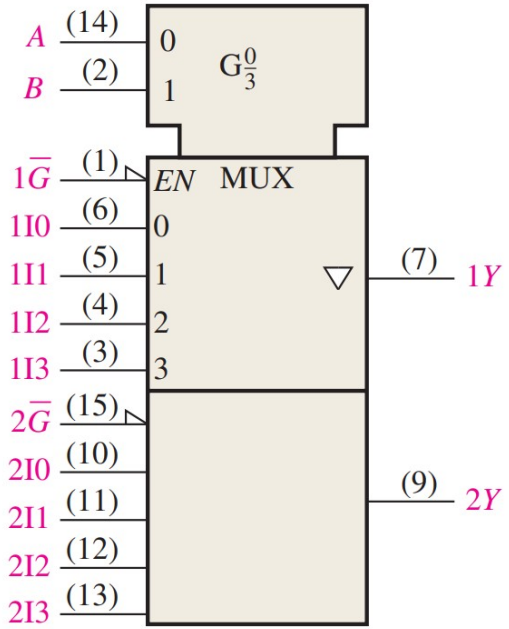


Real Chips – 74HC153

- Dual 4-input Ml
- Now with enabl



(a) Pin diagram

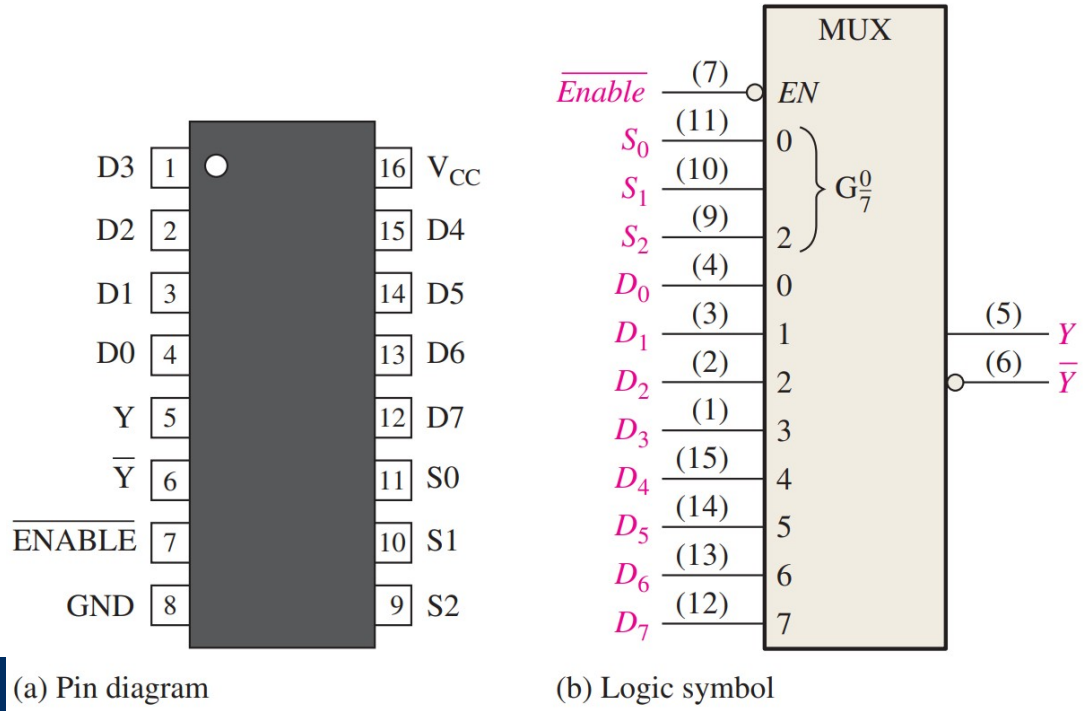


(b) Logic symbol

The 74HC153 dual four-input data selector/multiplexer.

Real Chips – 74HC151

- 8-input MUX



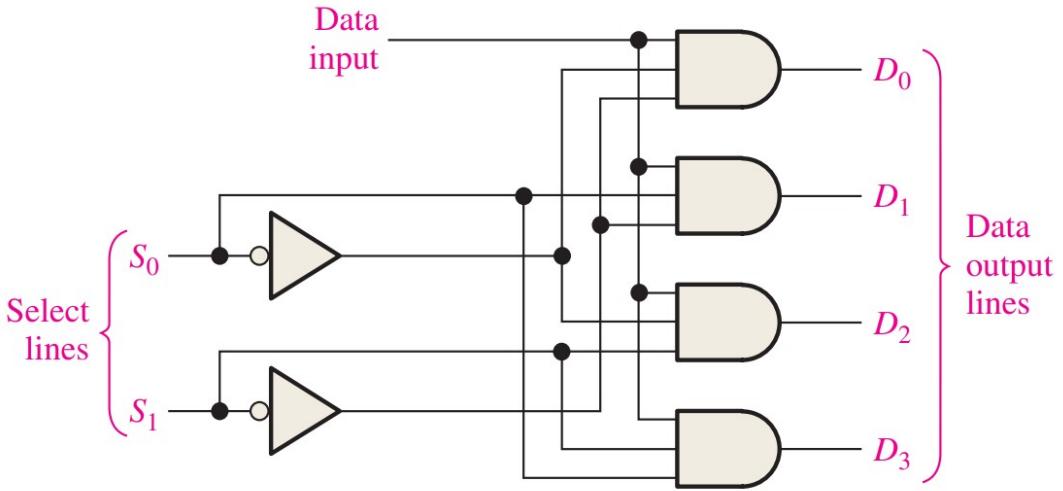
The 74HC151 eight-input data selector/multiplexer.

Demultiplexer (DEMUX)

- Reverse of MUX
  - n selection inputs and one data input
  - 2^n data outputs
  - n selects pins tell which of 2^n the input goes onto

DEMUX

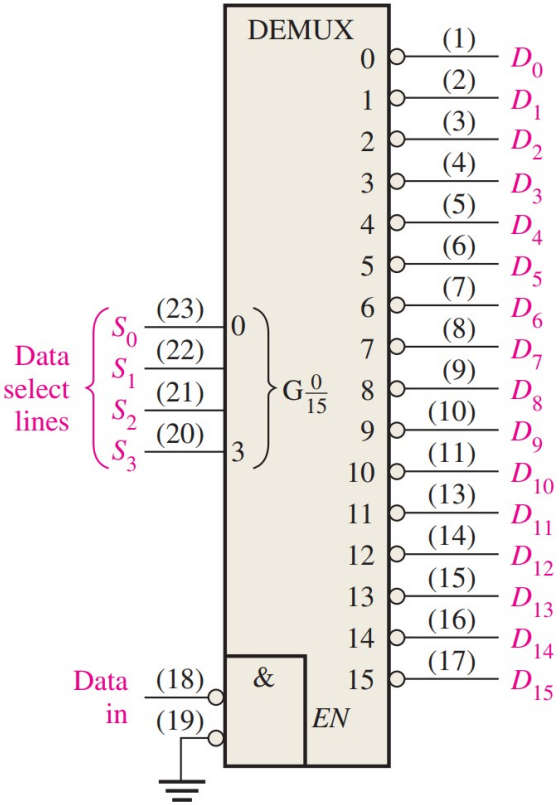
S1	S0	D3	D2	D1	D0
0	0	0	0	0	Input
0	1	0	0	Input	0
1	0	0	Input	0	0
1	1	Input	0	0	0



Decoder as DEMUX

- One CS line becomes data input

Active low example



DEMUX Applications

- Like a chip select to pick what subsystem

  to communicate with
  - Computer to projector or monitor
- Memory addressing and comms
- Serial to parallel conversion

Parity

- A form of error checking
- Even parity - # of 1’s in a transmission is even
  - Sum of all bits is 0
- Odd parity - # of 1’s in a transmission is odd
  - Sum of all bits is 1

Examples

- Even or odd parity?
- 1000 0100
- 1010 0101
- 1110 0000
- 0110 0100
- 0110 1010

Parity for Error Checking

- One less data bit than packet size
  - Replaced with parity bit
  - Packets without parity are discarded
    - Packet can be any size
    - Larger packets should have more robust error checking

Communication w/Parity

- One form of parity must be chosen
- Generate parity bits to ensure the chosen form
- Parity check passes if received data has the chosen form, otherwise error occurred

Examples

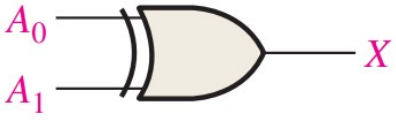
- Assume even parity communication
  - Find the parity bit x
- 1000 010x
- 1010 010x
- 1110 000x
- 0110 010x
- 0110 101x

Examples

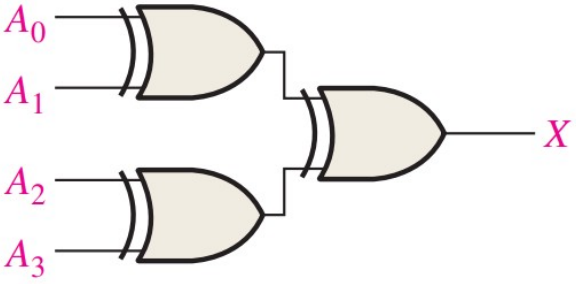
- Assume even parity communication
  - Is there a packet error?
- 1000 0101
- 1010 0100
- 1110 0001
- 0110 0101
- 0110 1011

Generating Parity

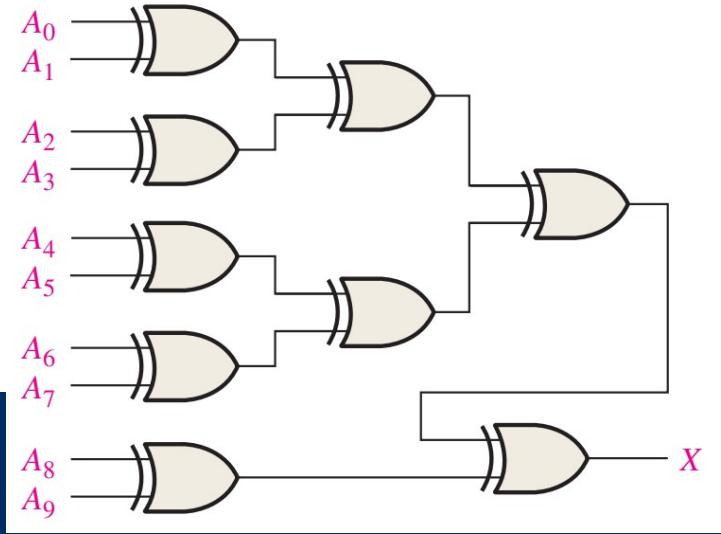
- Sum all bits to get even parity bit
- Sum and NOT to get odd parity



Summing of two bits



Summing of four bits



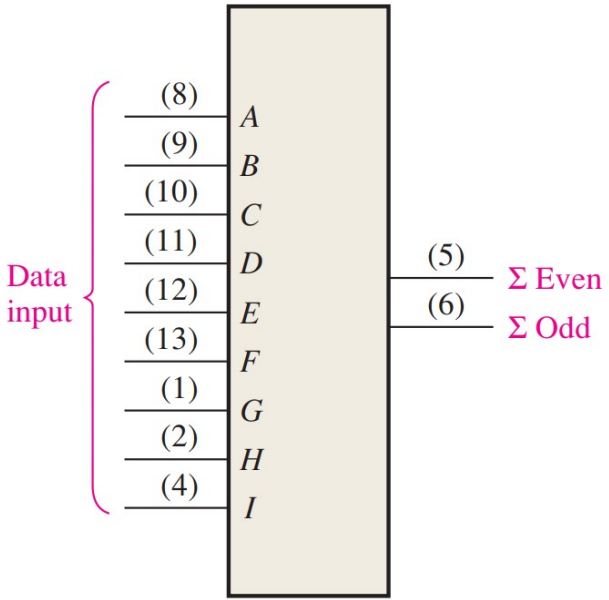
Checking Parity

- Same as generating, but with an extra bit
  - Use opposite parity to indicate an error

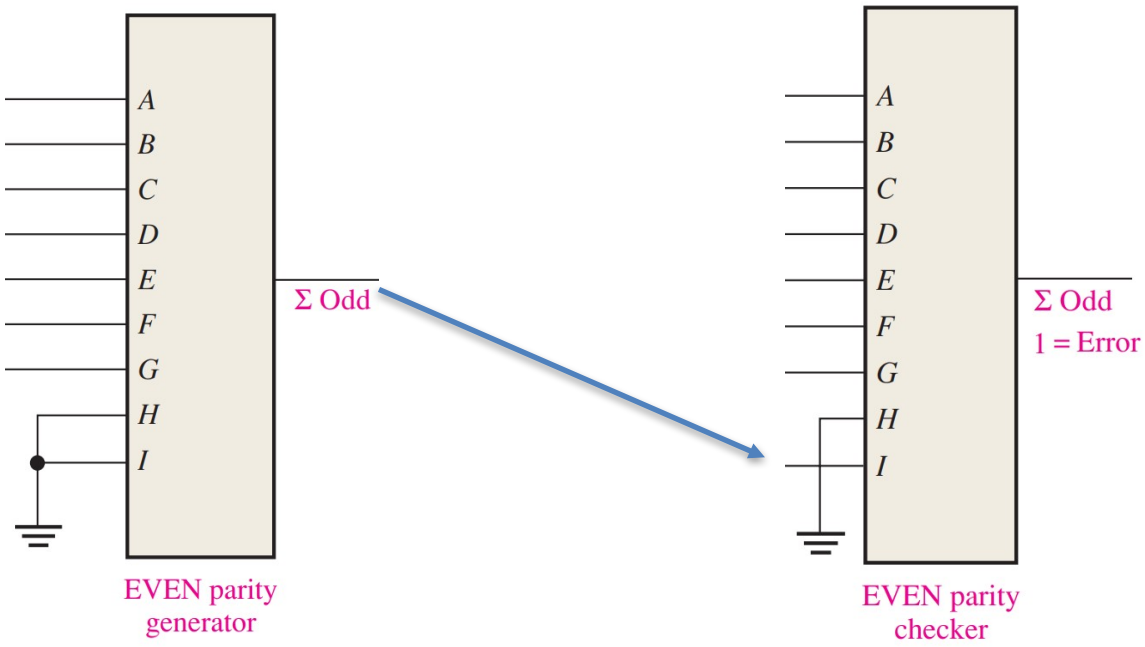
Real Chips – 74HC280

- 9-bit parity generator/chec

Number of Inputs *A–I* that Are High	Outputs Σ Even	Σ Odd
0, 2, 4, 6, 8	H	L
1, 3, 5, 7, 9	L	H



Use


A
B
C
D
E
F
G
H
I
Σ Odd
EVEN parity generator
A
B
C
D
E
F
G
H
I
Σ Odd
1 = Error
EVEN parity checker

Reading

- This lecture
  - Sections 6.8-6.10
- Next lecture
  - Sections 5.7, 6.11, 7.1