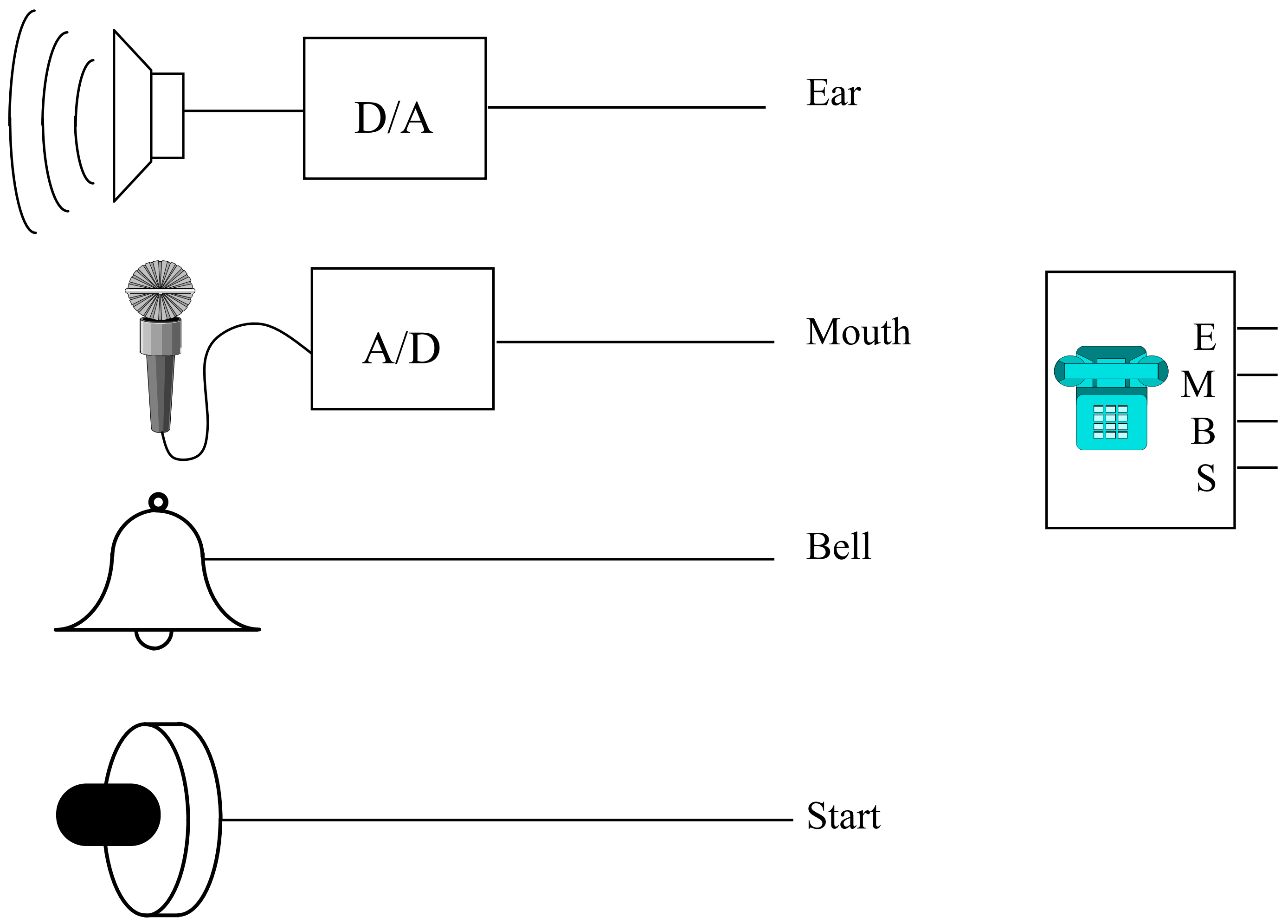
Basic Circuit Elements

- Readings: 4-4.1.1, 4.2, 4.3-4.3.2
- Standard TTL Small-Scale Integration:
  1 chip = 2-8 gates
  - Requires numerous chips to build interesting circuits
- Alternative: Complex chips for standard functions
  - Single chip that performs very complex computations
- Multiplexer/Decoder/Encoder: Standard routing elements for interconnections
- FPGAs: Programmable for arbitrary functions

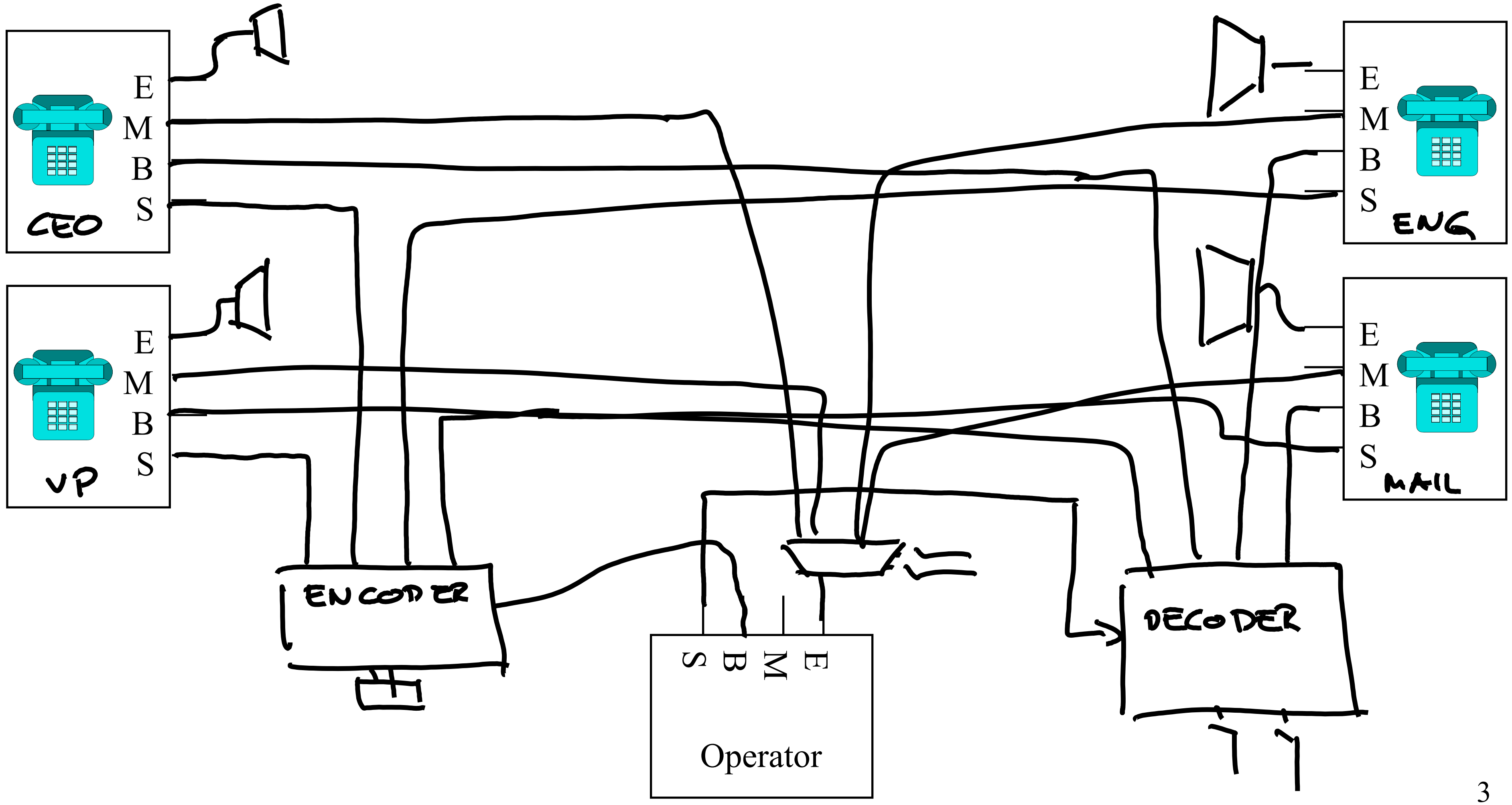
Design Example: Basic Telephone System

- Put together a simple telephone system



Basic Telephone System (cont.)

- Multiple subscribers, one operator.
    - Operator controls all connections

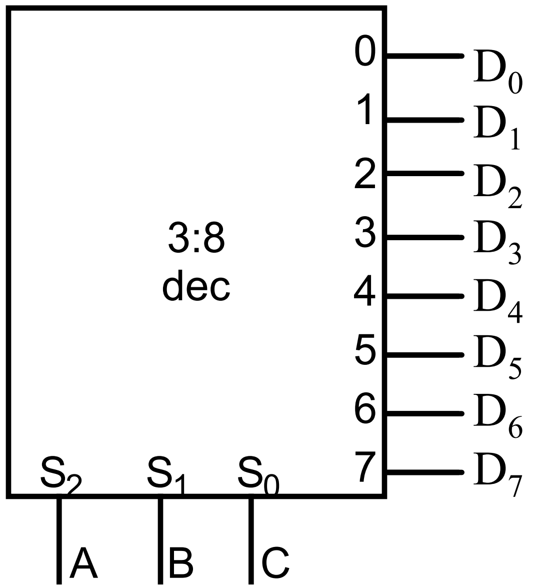


Standard Circuit Elements

- Develop implementations of important “Building Blocks”
  - Used in Networks, Computers, Stereos, etc.
- Multiplexer: Combine N sources onto 1 wire
- Encoder: Determine which input is active
- Decoder: Convert binary to one-of-N wires

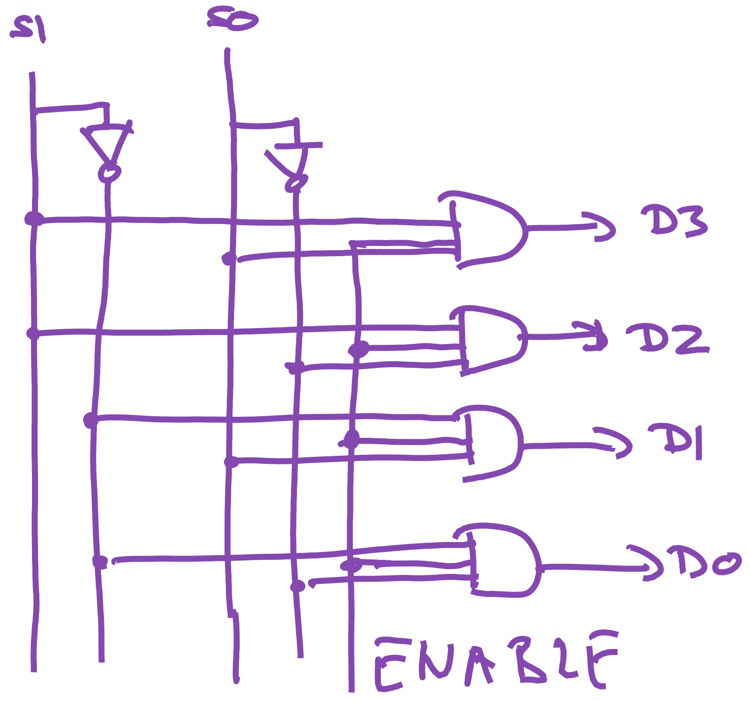
Decoders

- Used to select one of 2^N outputs based on N input bits
- Input: N bits; output: 2^N outputs -- only *one* is active
- A decoder that has *n* inputs and *m* outputs is referred to as an *n* x *m, N:M,* or *n-to-m* decoder
- Example: 3-to-8 decoder



Decoder Implementation

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



Enabled Decoder Implementation

❖ Active High enable

En	S1	S0	D3	D2	D
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			

Enabled Decoders in Verilog

```
module enDecoder2_4 (out, in, enable);
  output reg [3:0] out;
  input      [1:0] in;
  input            enable;

  always @(*) begin
     if (enable)
        case (in)
        2'b 00 : out= 4'b 0001;
        2'b 0 1 : out = 4'b 0010;
          :

        endcase
     else
     out = 4'b0000;


endmodule
```

Encoders

- Performs the inverse operation of decoders
- Input: 2^N or less lines -- only 1 is asserted at any given time
- Output: N output lines
- Function: the output is the binary representation of the ID of the input line that is asserted

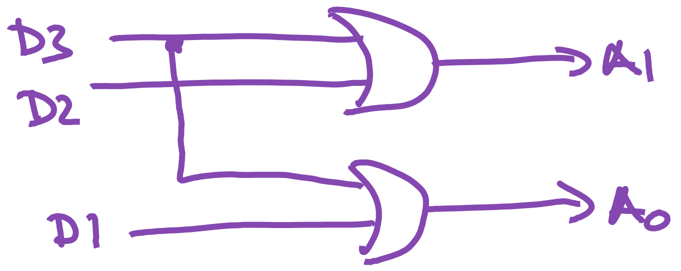
Encoder Implementation

- 4:2 Encoder

D3	D2	D1	D0	A1	A0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

$A_1 = D_3 + D_2$

$A_0 = D_3 + D_1$



1) WHEN NO ONE CALLS $\longrightarrow$ CODE 00

2) D2 AND D1 CALL $\longrightarrow$ CODE 11 (D3)

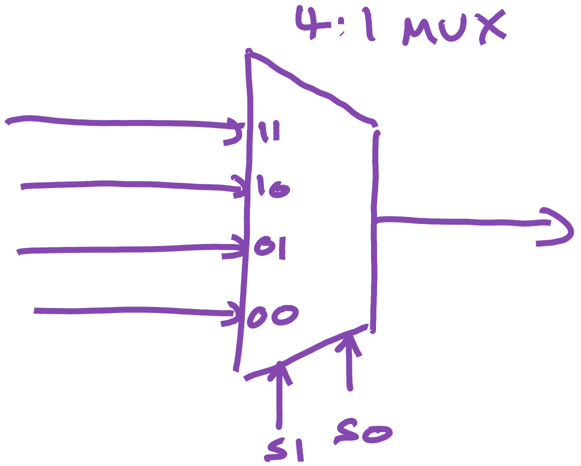
Priority Encoder

- Use priorities to resolve the problem of 2 or more input lines active at a time.
- One scheme: Highest ID active wins
- Also add an output to identify when at least 1 input active

D3	D2	D1	D0	A1	A0	VALID
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

Multiplexer

- An element that selects data from one of many input lines and directs it to a single output line
- Input: 2^N input lines and N selection lines
- Output: the data from *one* selected input line
- Multiplexer often abbreviated as MUX



Multiplexer Implementation

- 4:1 MUX

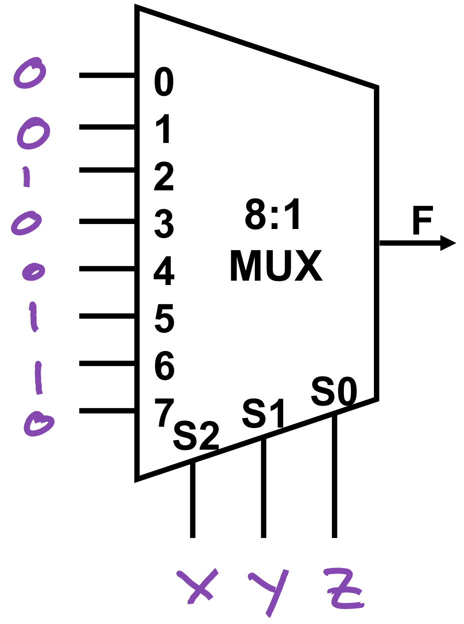
S1	S0	F
0	0	D0
0	1	D1
1	0	D2
1	1	D3

$$F = \overline{S1}\,\overline{S0}\,D0 + \overline{S1}\,S0\,D1 + S1\,\overline{S0}\,D2 + S1\,S0\,D3$$

Multiplexers in General Logic

- Implement $F = X\overline{Y}Z + Y\overline{Z}$ with a 8:1 MUX

X	Y	Z	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0



0
0
1
0
0
1
1
0
0
1
2
3
4
5
6
7
8:1
MUX
F
S2
S1
S0
X
Y
Z