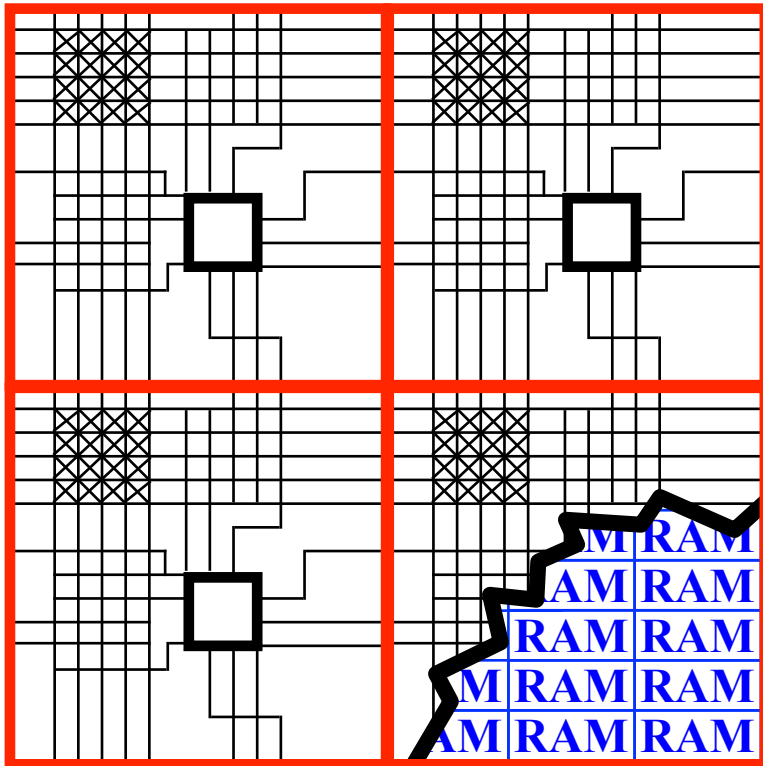


# Field Programmable Gate Arrays (FPGAs)

❖ Readings: B.6-B.6.5



**Logic cells imbedded in a general routing structure**



**Logic cells usually contain:**

- **6-input Boolean function calculator**
- **Flip-flop (1-bit memory)**

**All features electronically (re)programmable**

# Using an FPGA

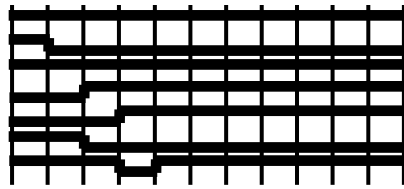
```
// Verilog code for 2-input multiplexer
module AOI (F, A, B, C, D);
  output F;
  input A, B, C, D;
  assign F = ~(A & B) | (C & D);
endmodule

module MUX2 (V, SEL, I, J); //
  2:1 multiplexer
  output V;
  input SEL, I, J;
  wire SELB, VB;
  not G1 (SELB, SEL);
  AOI G2 (VB, I, SEL, SELB, J);
  not G3 (V, VB);
endmodule
```

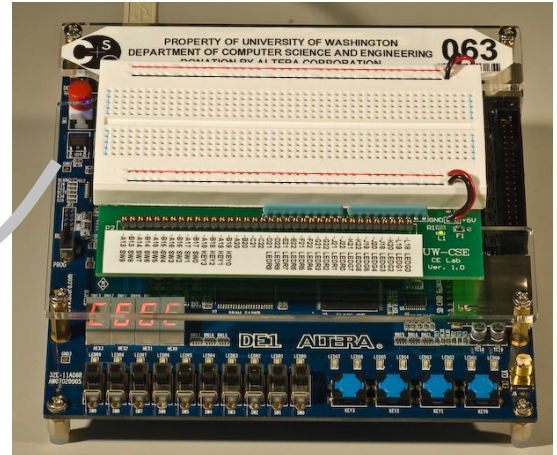
Verilog



FPGA  
CAD  
Tools



Simulation



```
00101010001010010
10010010010011000
10101000101011000
10101001010010101
00010110001001010
10101001111001001
01000010101001010
10010010000101010
10100101010010100
01010110101001010
01010010100101001
```

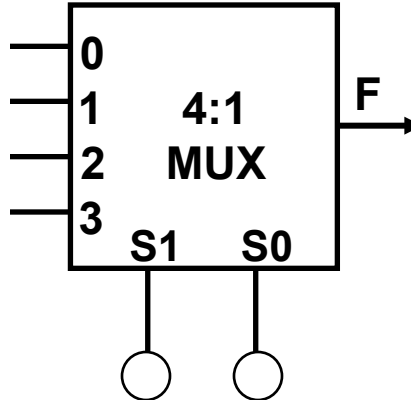
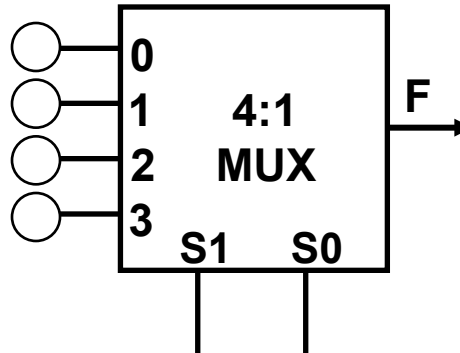
Bitstream

# FPGA Programming

---

```
00101010001010010
10010010010011000
10101000101011000
10101001010010101
00010110001001010
10101001111001001
01000010101001010
10010010000101010
10100101010010100
01010110101001010
01010010100101001
```

Bitstream

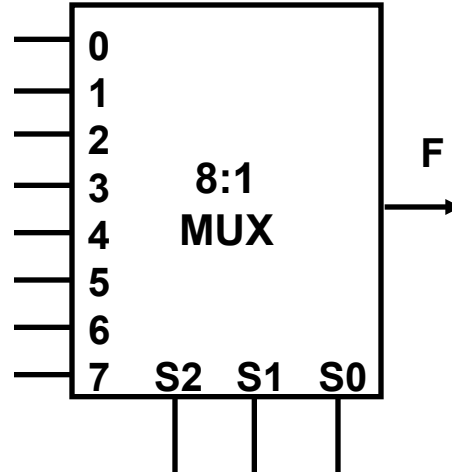


○ = 1 memory cell (stores 1 bit of info)

# FPGA Combinational Logic

---

- How can we use Muxes and Programming bits to compute combinational binary function  $F(A,B,C)$ ?

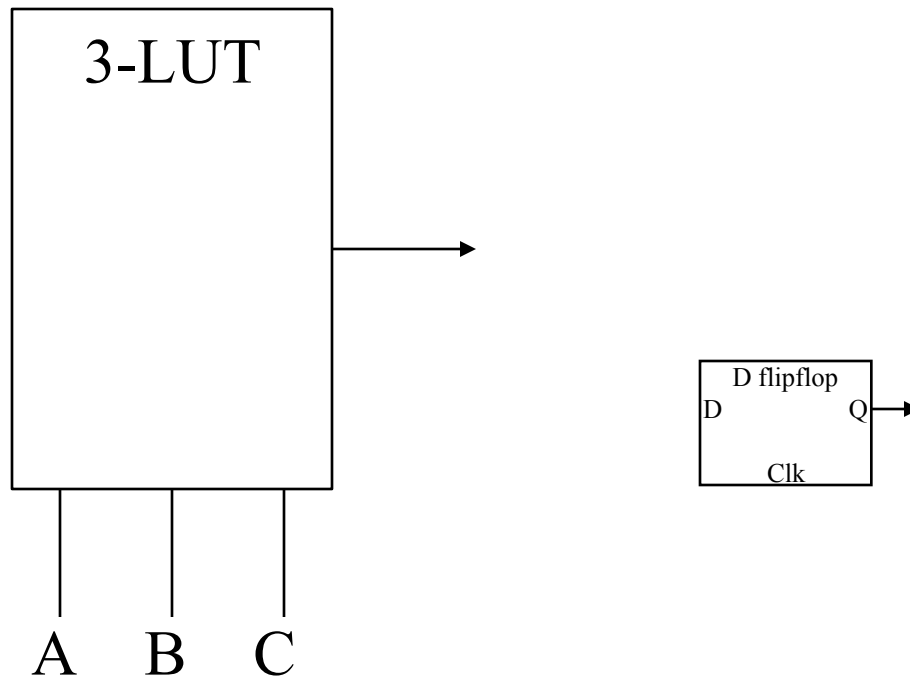


- Creates a “LUT” or lookup table.

# FPGA Sequential Logic

---

- How do we put DFF's onto LUT outputs only when we need them?

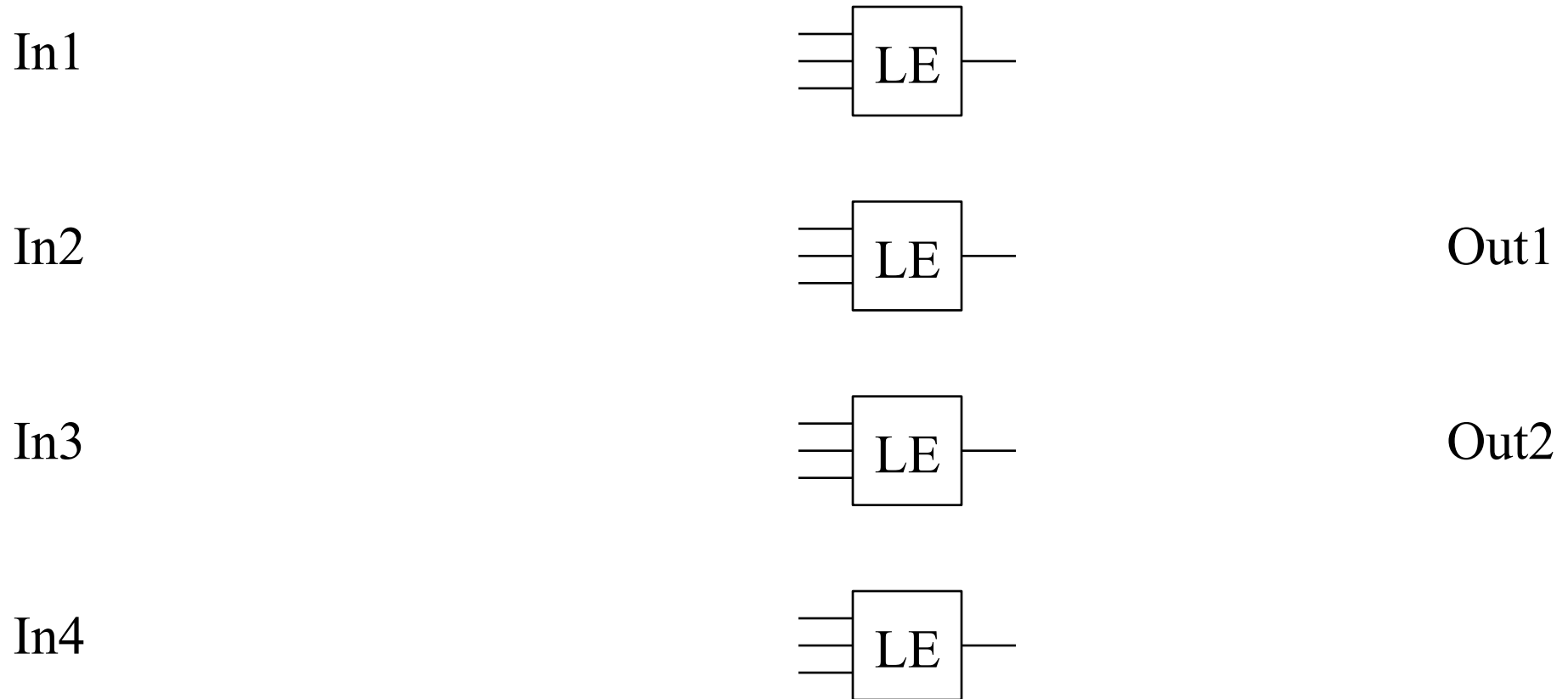


- Creates a “LE” or logic block

# FPGA Local Routing

---

- How do we combine LE's to build larger functions?

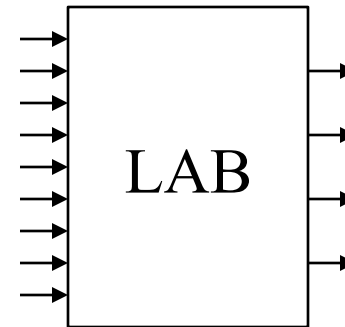
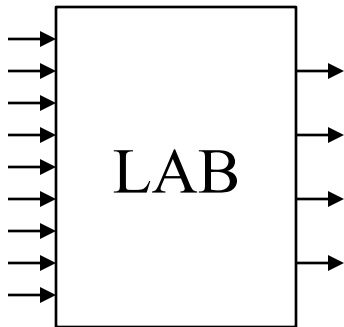
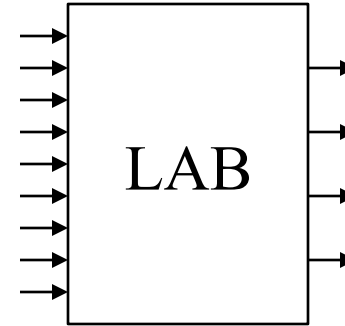
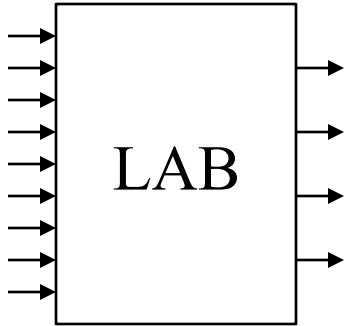


- This is an Altera “LAB”.

# FPGA Global Routing

---

- Can't do all-to-all/crossbar routing, so what?



# FPGA CAD

## ■ CAD = “Computer-Aided Design”

```
// Verilog code for 2-input
multiplexer
module AOI (F, A, B, C, D);
  output F;
  input A, B, C, D;
  assign F = ~(A & B) | (C & D);
endmodule

module MIX2 (V, SEL, I, J); //
  2:1 multiplexer
  output V;
  input SEL, I, J;
  wire SELB, VB;

  not G1 (SELB, SEL);
  AOI G2 (VB, I, SEL, SELB, J);
  not G3 (V, VB);
endmodule
```

Verilog



FPGA  
CAD  
Tools



```
00101010001010010
10010010010011000
101010000101011000
10101001010010101
00010110001001010
10101001111001001
01000010101001010
10010010000101010
10100101010010100
01010110101001010
01010010100101001
```

Bitstream

- Tech Mapping: Convert Verilog to LUTs
- Placement: Assign LUTs to specific locations
- Routing: Wire inputs to outputs
- Bitstream Generation: Convert mapping to bits