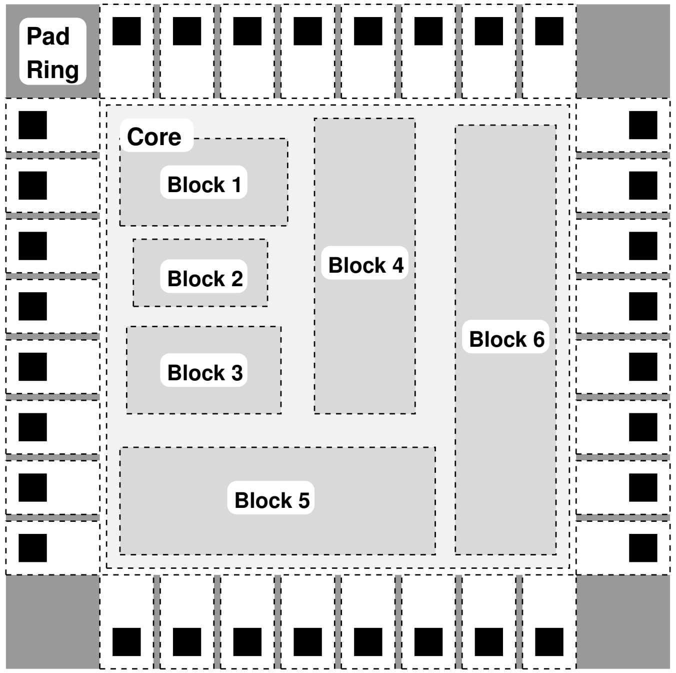
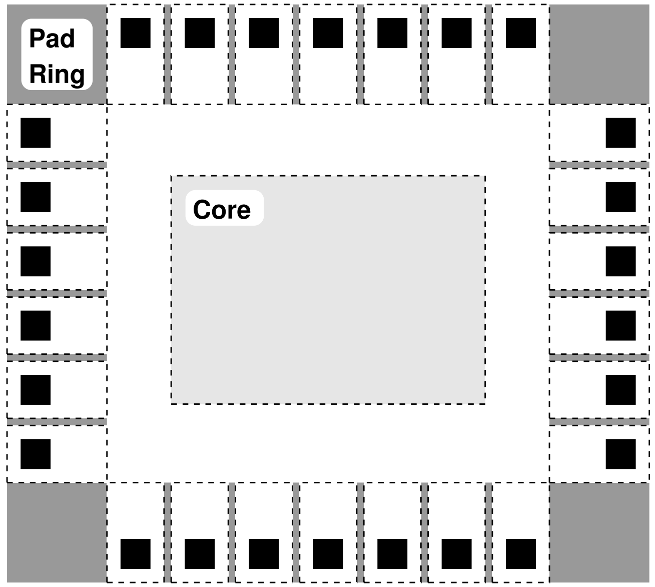
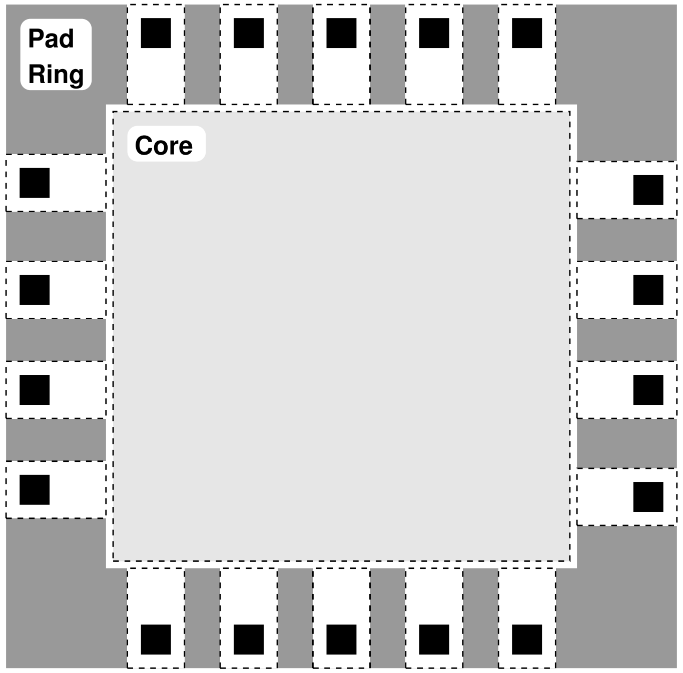
Pad Ring and Floor Planning



- The core of the chip (made up of one or more top level blocks) is surrounded by a ring of pads.
- The design of the blocks and the arrangement of blocks and pads can significantly affect the overall chip area (and hence the cost/yield).

Pad Ring





Pad Limited: small core and/or many pads

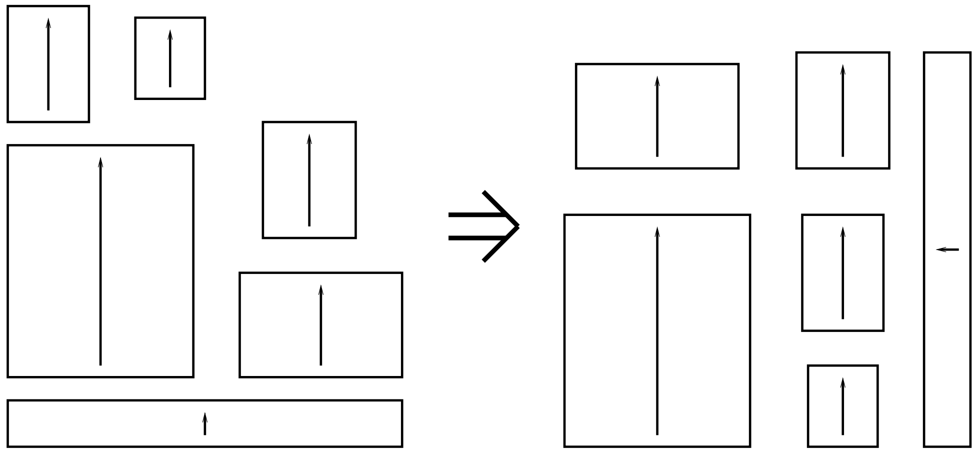
minimum pad to pad distance – gaps around core

Core Limited: large core and/or few pads

gaps between pads[1]

[1]these gaps will be filled with special filler cells

Floor Planning



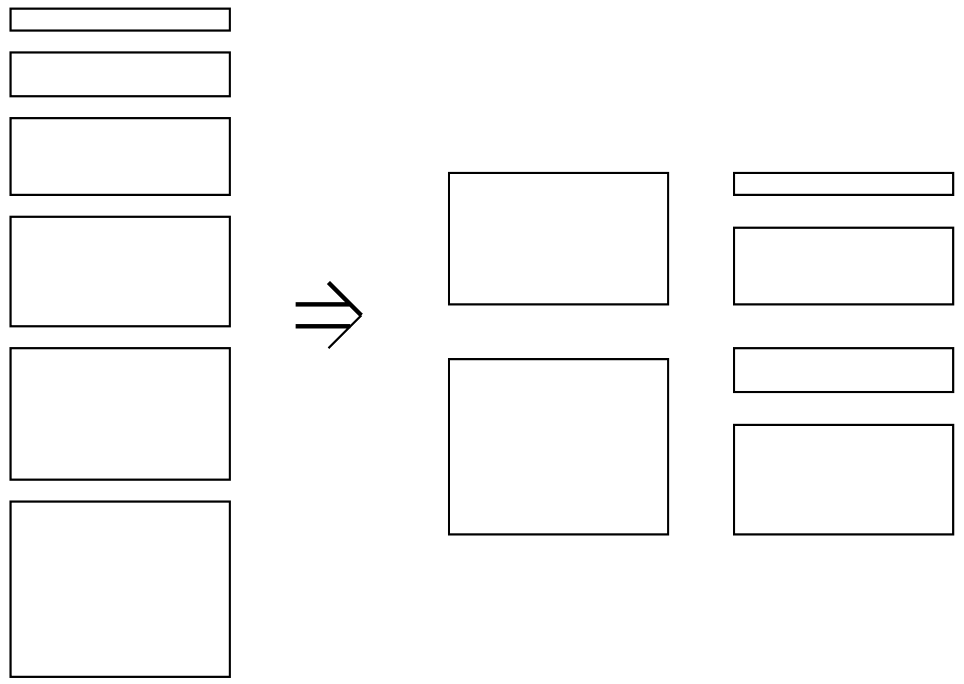
- Re-arrange and re-orient blocks to:
  - create a minimum number of major routing channels[2]
  - reduce block to block and block to pad routing

At top of the hierarchy, chips should be near square, other constraints exist at lower levels.

[2]for multi layer metal processes (≈ 5 metal layers or more) it should be possible to route over the blocks allowing closer placement

Block Design for easy Floor Planning

- Block shape



  Where blocks share a common width, efficient placement is much easier.

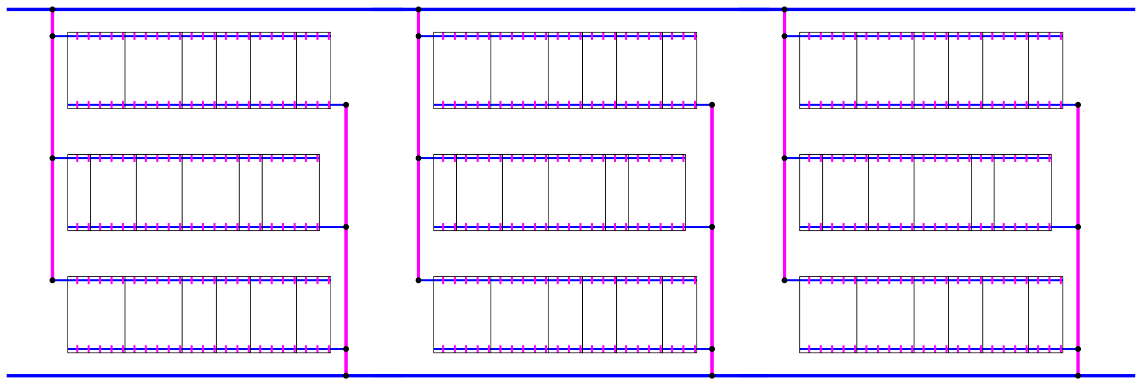
- Block ports

  If possible arrange the ports on a block for ease of routing to pads and other blocks.

Floor Planning for Standard Cell Layout

Automatic layout:

- Flatten hierarchy.
- Placement is controlled by algorithms designed to minmize routing.
- Aspect ratio easy to control, also control number of columns and rows.



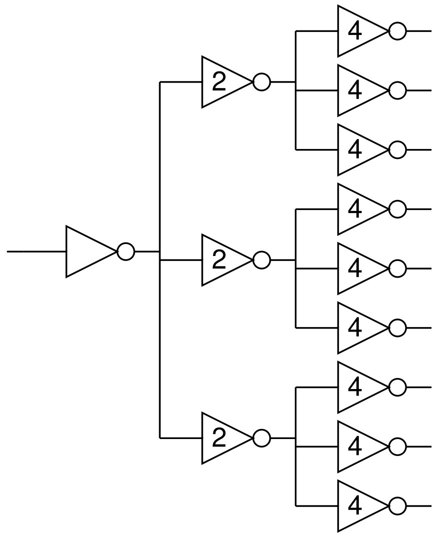
Manual layout:

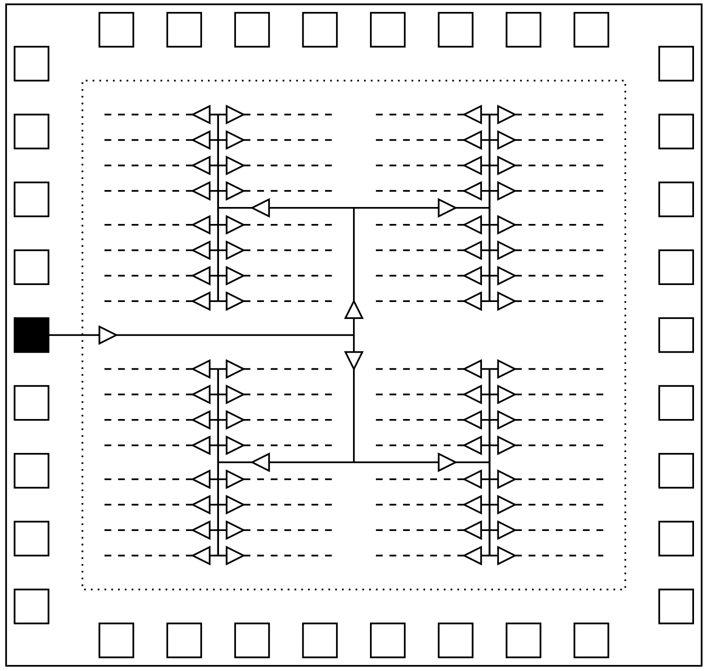
- Placement based on layout hierarchy (essential for managing complexity).
- Aspect ratio and port position must be considered early as there is seldom time for iteration.

Global Routing

Route critical signals first.

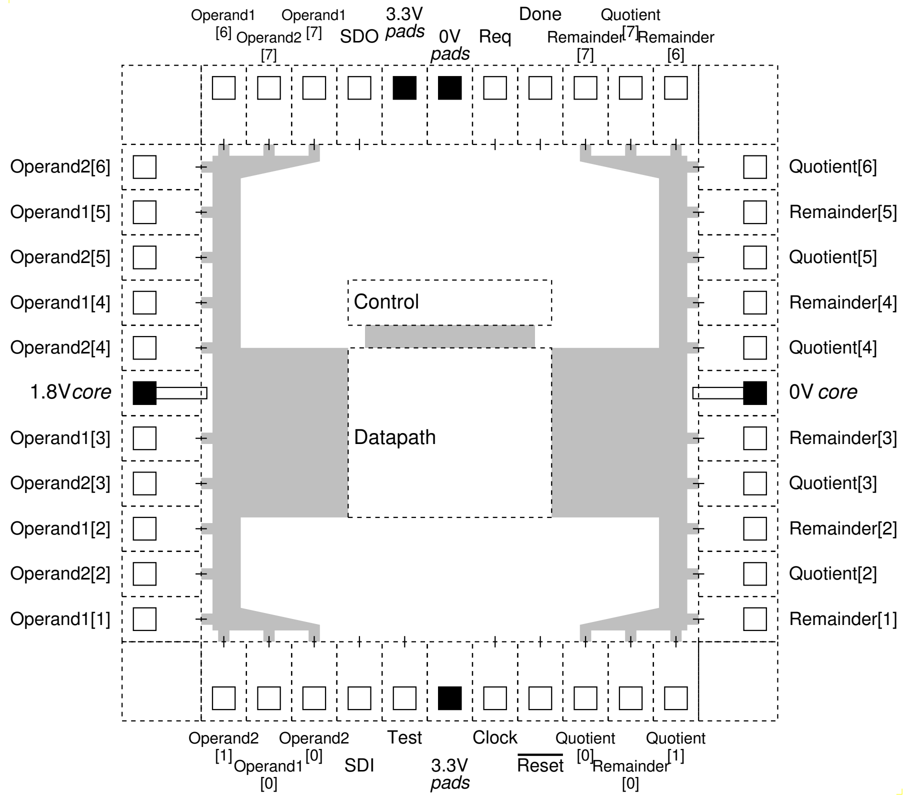
- Buffer global and time critical signals.
- Clock distribution should be arranged to avoid skew across the chip[3].





[3]buffering may actually increase delays while reducing skew

VLSI – Pad Ring and Floor Planning



- Pad ring pre-defined[a]

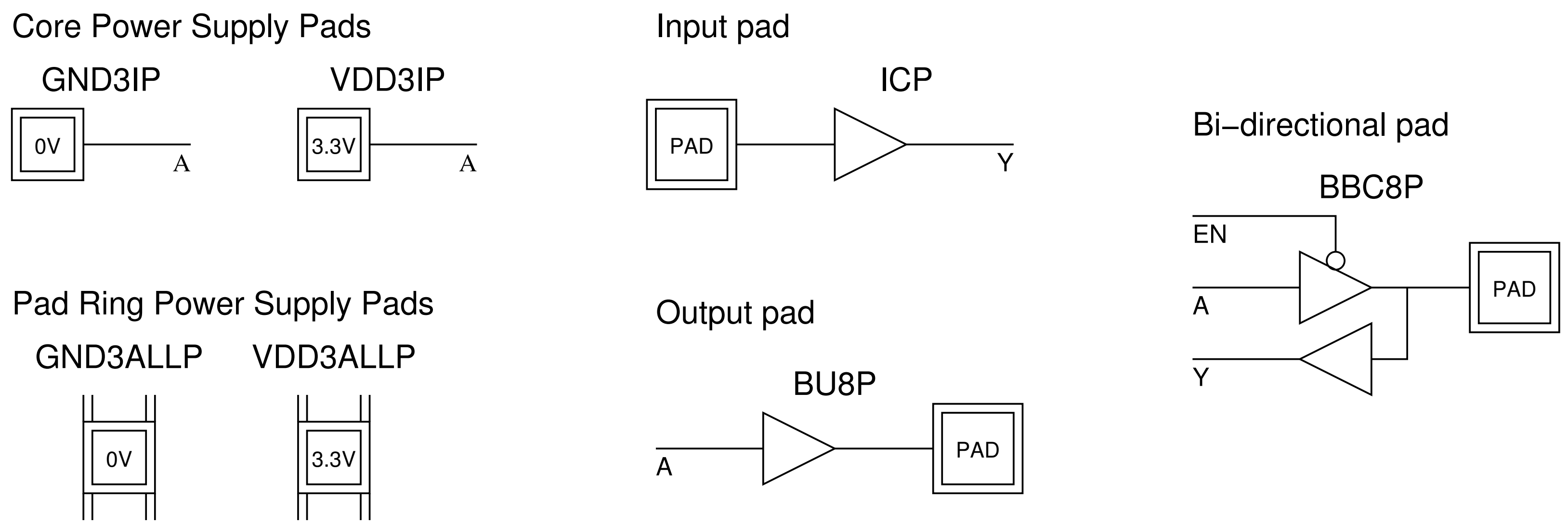
```
create_pad_ring
  -divider
  <xsize> <ysize>
```

- Two blocks in core
  - Bitslice Datapath
  - Synthesized Control
- Pad limited
- Clock distribution built in to cell library

[a]design blocks to reduce routing since pads can't be moved

Datapath will be designed and placed to permit easy wiring of Operand and Quotient+Remainder buses to left and right hand pads. Control will be designed and placed to permit easy wiring of control signals to the datapath.

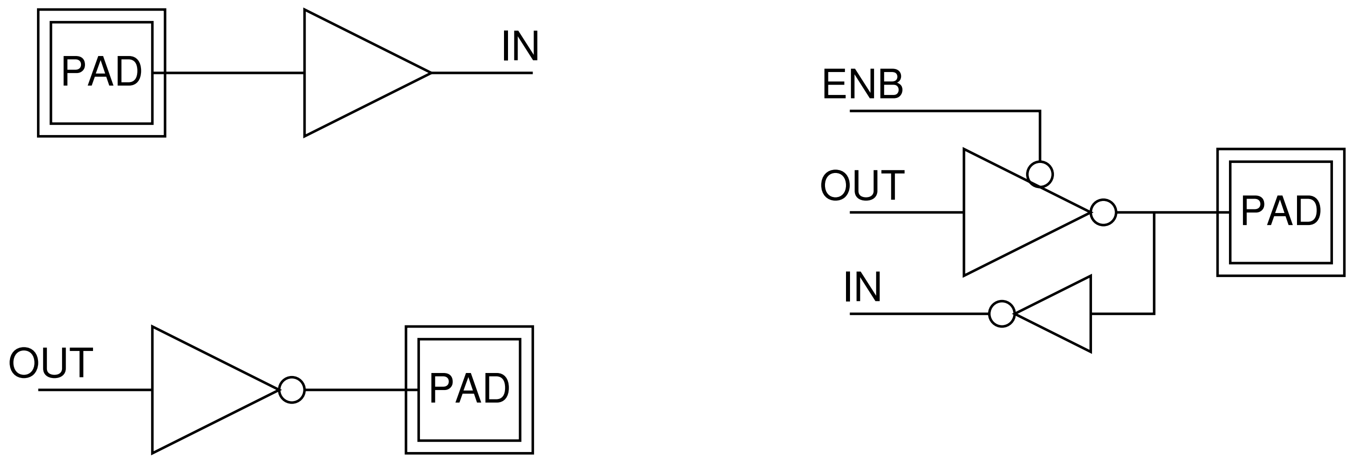
VLSI – AMS 0.35μm CMOS Pads



- Large buffers on output pads allow for drive of very large external loads.
- Separate ”dirty power” supply pads are provided for the main pad drive transistors to reduce switching noise in the core.
- Bi-directional pads require three connections to the core.

Input / Output

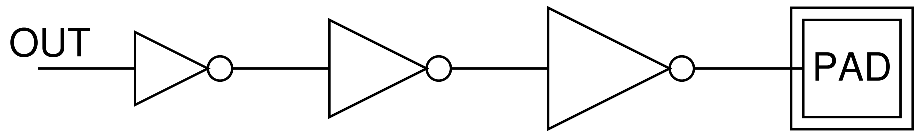
- I/O Pads



  - A brief look at a selection of simple digital CMOS I/O pads

Output Pads

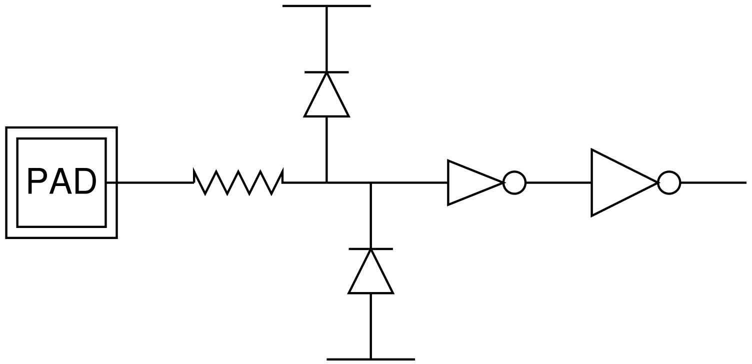
- Output pad driver



  - ratioed inverters are used to provide appropriate drive capability
  - final drive transistors are carefully designed to avoid latch-up
  - pad rings are frequently powered separately (dirty power) to confine switching noise

Input Pads

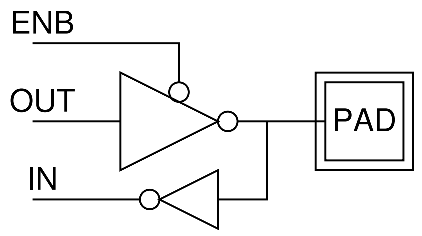
- Input protection

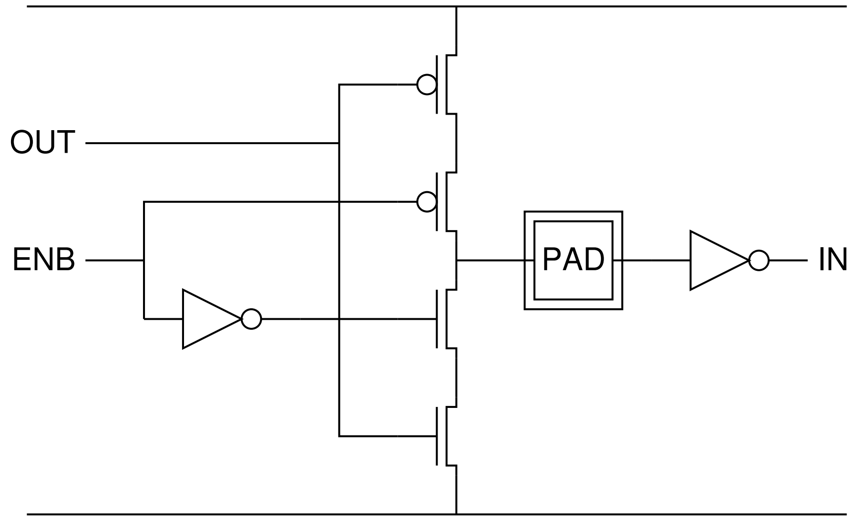


  - must protect floating transistor gates from permanent damage via electro-static discharge

Bidirectional Pads

- Simple bidirectional pad



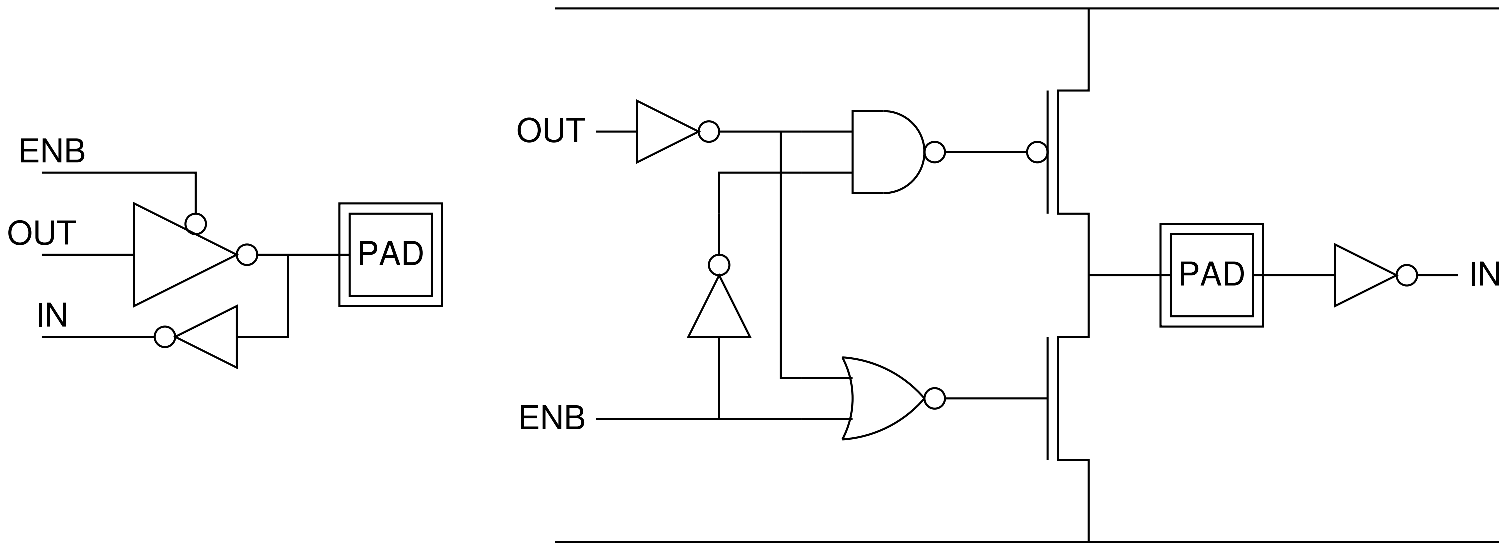


  - bidirectional pad is a tristate inverter output driver combined with an input pad[4]
  - even when `IN` and `OUT` are connected internally, we need buffering and an enable control signal

[4]note input protection is not shown here

Bidirectional Pads

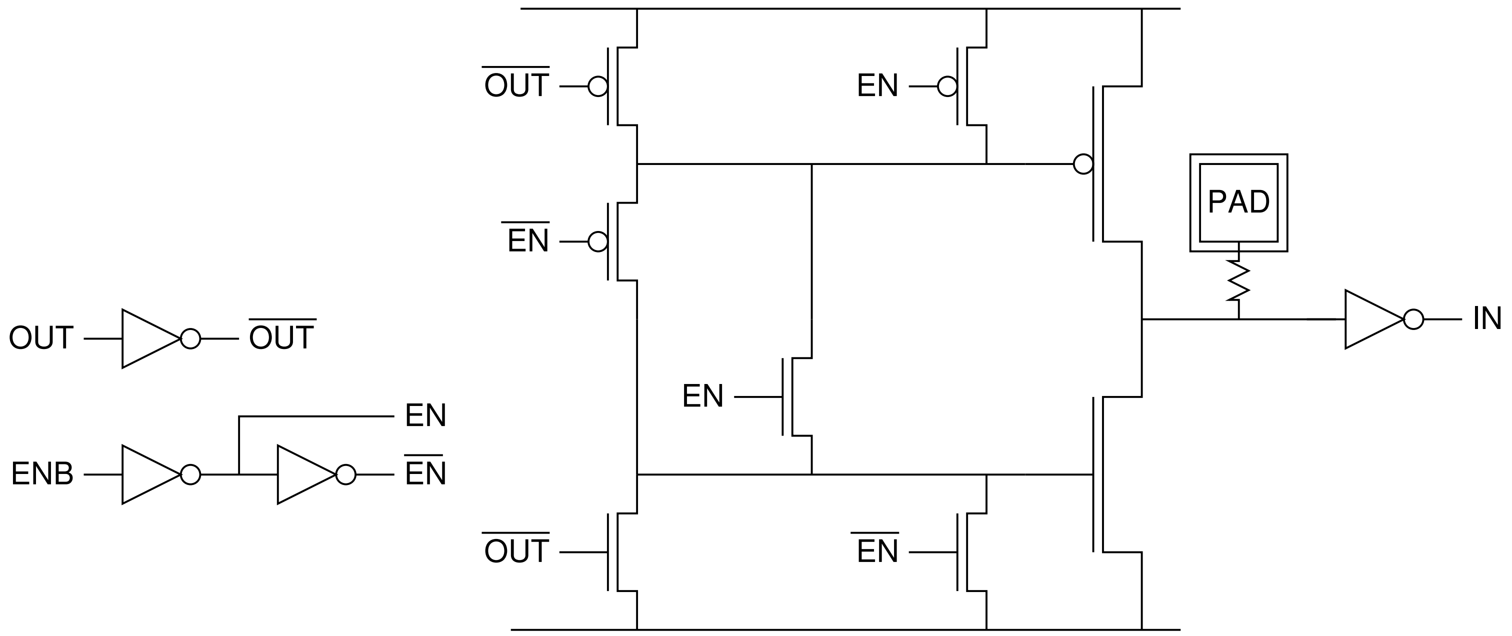
- Bidirectional pad with increased drive capability



  - redesign to avoid series output transistors

Bidirectional Pads

- Advanced bidirectional pad design



  - logic gates are merged
  - output transistors act as diodes when not enabled
  - low value diffusion resistor completes input protection circuit