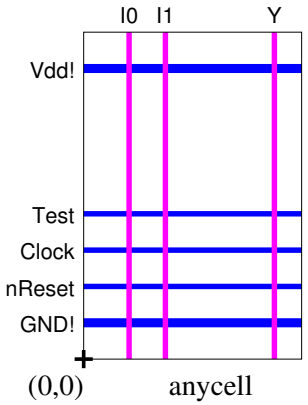
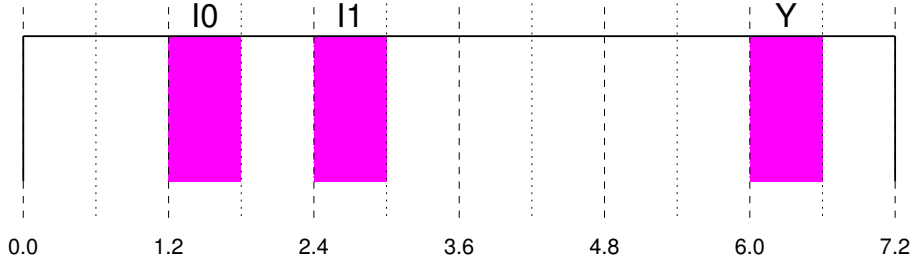
DICD Coursework

Cell Library Design

- Teams of four or five.
- Sixteen gates to be designed:
  - Scannable D-type flip flop
  - Scannable Register
  - Full Adder, Half Adder
  - 2 input Multiplexer
  - 2 input NAND/NOR/XOR/AND/OR
  - 3 input NAND/NOR, 4 input NAND
  - Inverter, Buffer, Tristate Buffer
- Additionally five support cells to be designed:
  - Tie High and Tie Low cells
  - Row Crosser cell
  - Left and Right End of Row cells

- Technology: On Semi 0.35μm unified CMOS (use `magic -T c035u`)
- All transistors fixed size: W_P = 2.4μm , W_N = 1.5μm , L_P = L_N = 0.35μm
- Horizontal i/o in metal1 (including power/ground and global signals)
- Vertical i/o in metal2 (aligned on 1.2μm grid)
- Power rails: 1.25μm (all other dimensions kept to a minimum)
- Taps arranged along power rails - joined by continuous nohmic/pohmic
- Cell aligned to origin (0,0) - cell width divisible by 1.2μm
- Gate Matrix construction (where possible)





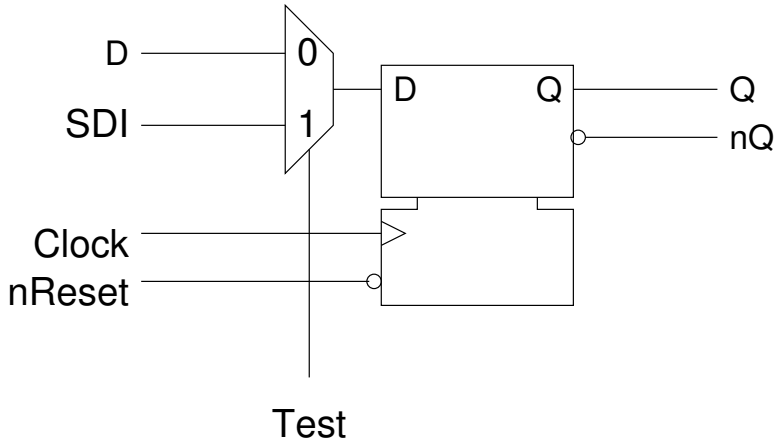
Alignment of vertical i/o to 1.2 micron grid
(to support magic auto−router)

Scannable D-type flip-flop

- triggered on rising edge of clock
- with asynchronous active low reset

Scan path support

- In normal operation the D-type receives data on its D input but when the Test signal is 1, data is received on the *scan data in* (SDI) input.

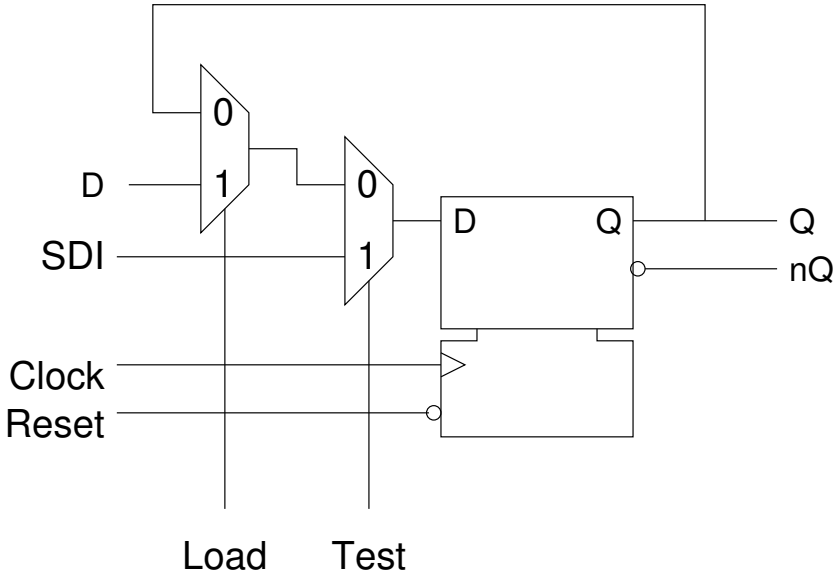


Scannable Register

- synchronous load signal allows for a new value to be loaded
- otherwise Q is fed back to the input

Scan path support

- Scan operation overrides register operation.

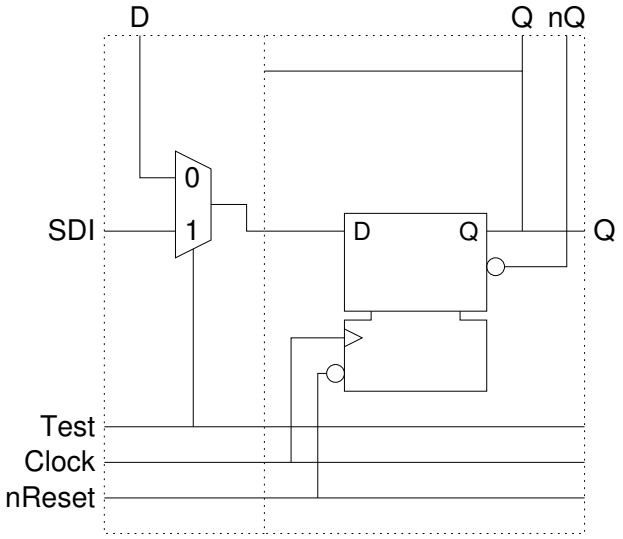


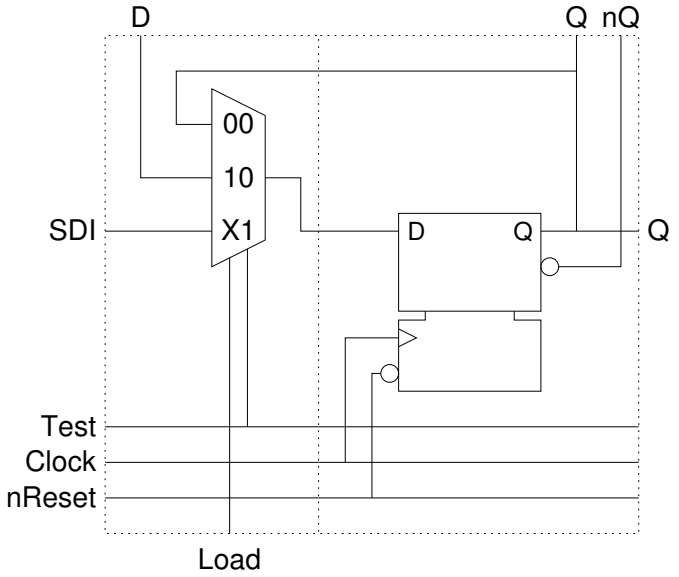
Re-use of cells

To avoid duplication of effort on the D-type, the actual cells designed will be:

- Raw D-type (no multiplexer)
- 2 and 3 input scan multiplexers

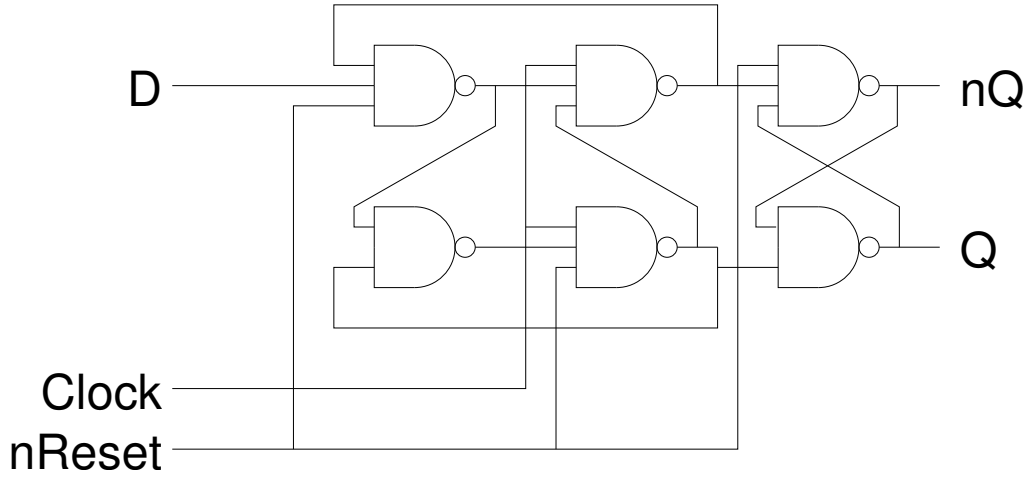
The Scannable D-type is constructed by butting the raw D-type and the 2 input scan multiplexer. The scan path is automatically joined when scannable gates are butted. The cell inputs and outputs are arranged to facilitate this butting:

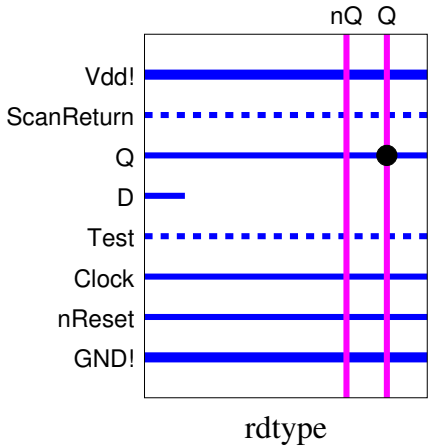




Raw D-Type

- 6 nand gate implementation
- horizontal i/o in metal1
- vertical i/o in metal2
- *order of signals may be customized by teams*

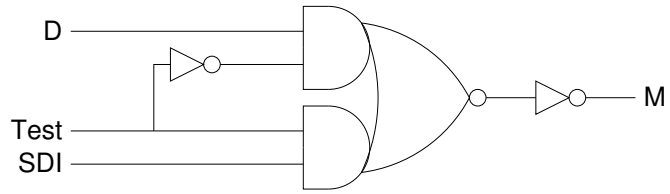


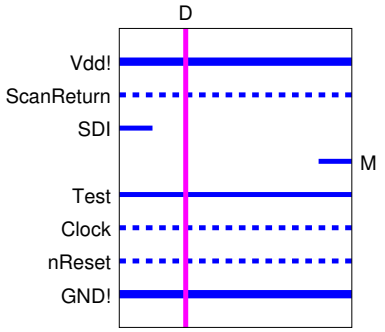


rdtype

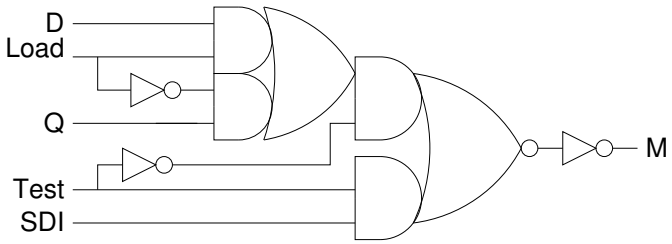
Scan Multiplexers

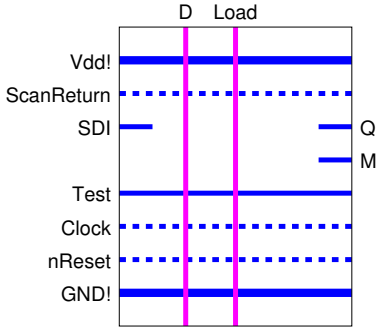
- compound gate implementation
- investigate Euler paths





smux2

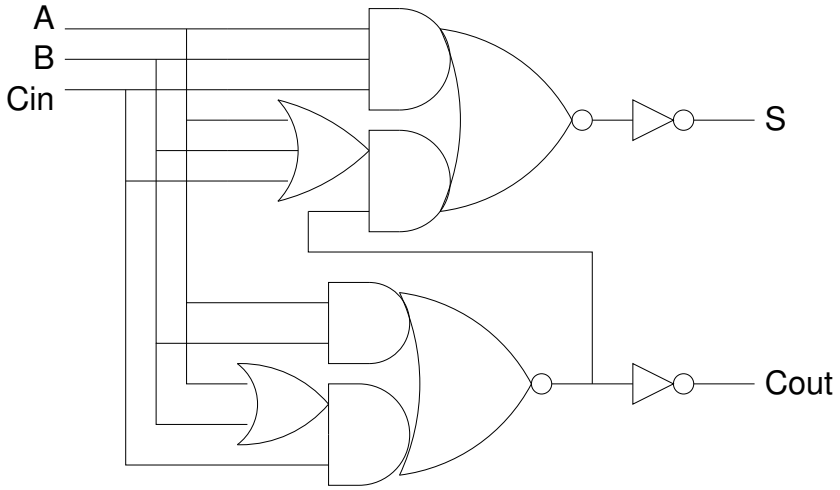


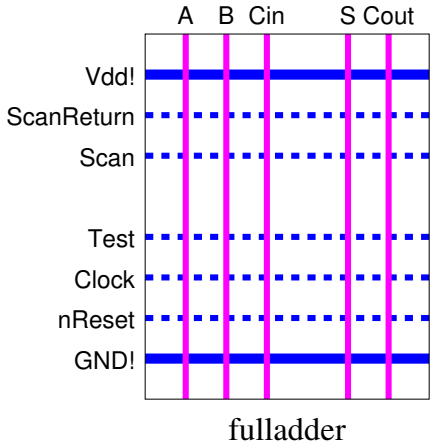


smux3

Full Adder

- compound gate implementation
- investigate Euler paths
- *Clock, nReset, Test, Scan and ScanReturn pass through without connection*

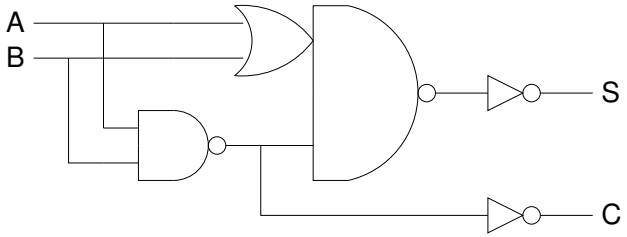


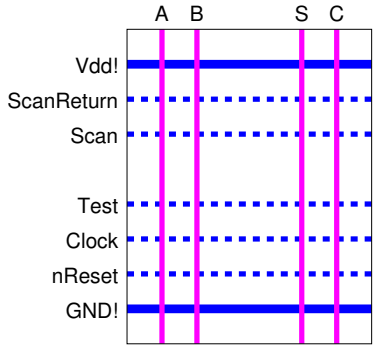


fulladder

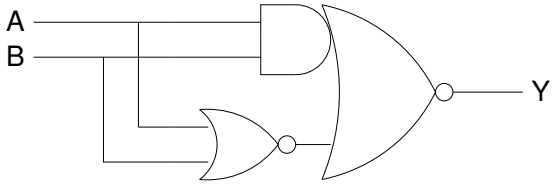
Half Adder and XOR

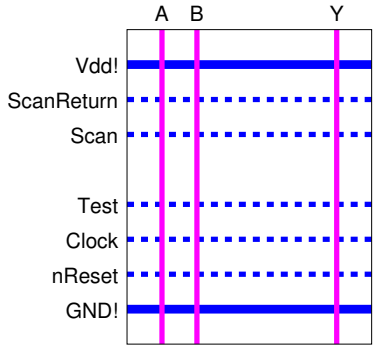
- compound gate implementation
- investigate Euler paths





halfadder



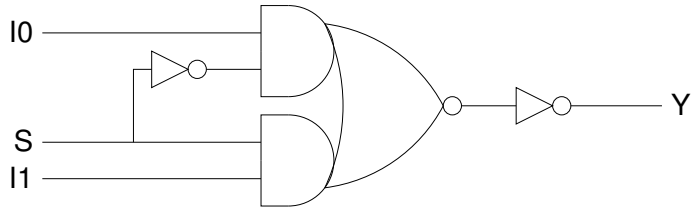


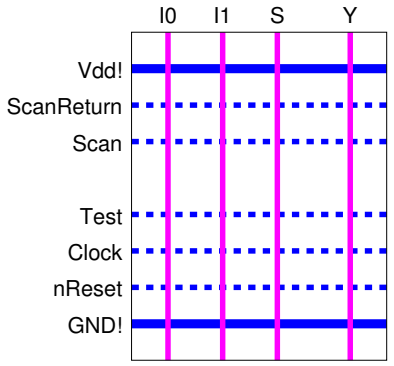
xor2

Standard 2 Input Multiplexer

This gate is a modified version of the **smux2** cell

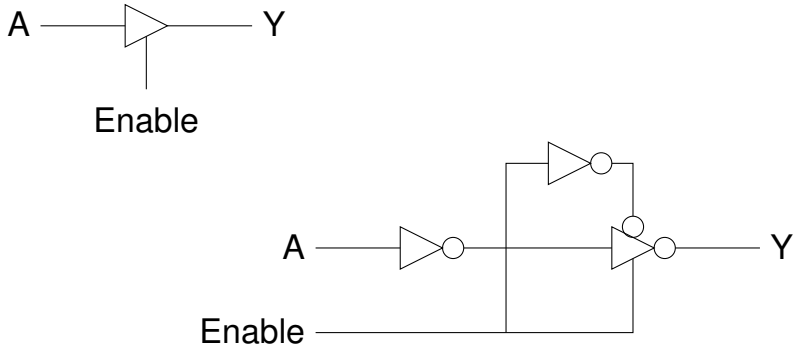
- vertical i/o (I0, I1, S, Y)
- Scan/Test pass through without connection

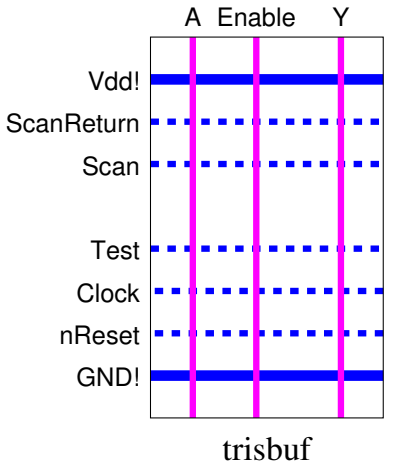




mux2

Tristate Buffer

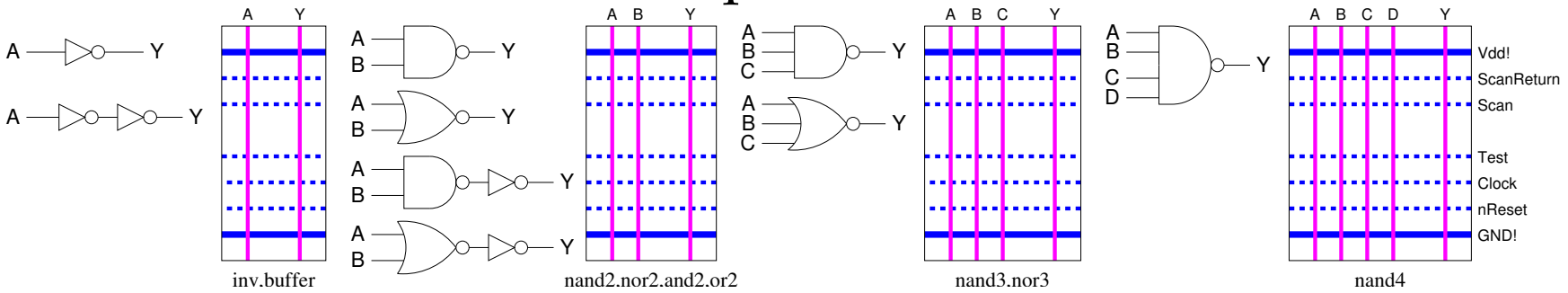




- trisbuf(Y, A, Enable)

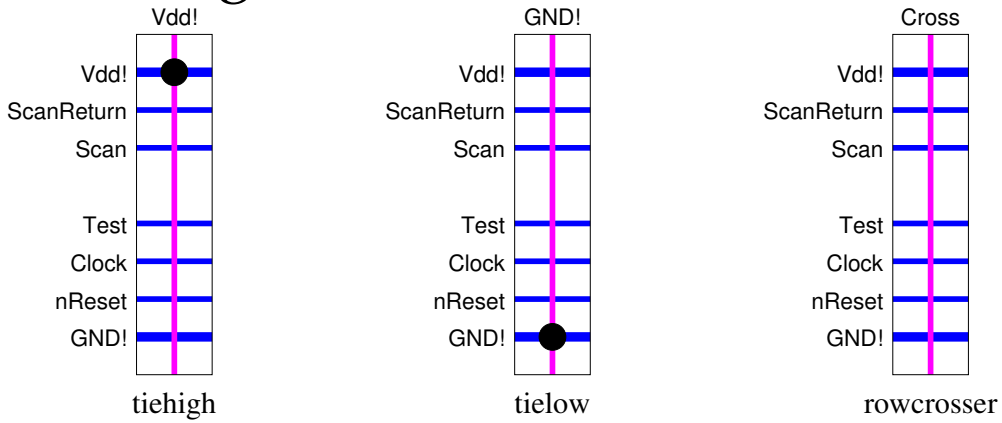
Cell Library Design

Simple Cells



- inv(Y, A)
- buffer(Y, A) - *use single Euler path for both transistors*
- nand2(Y, A, B)
- nor2(Y, A, B)
- and2(Y, A, B) - *use single Euler path for all three transistors*
- or2(Y, A, B) - *use single Euler path for all three transistors*
- nand3(Y, A, B, C)
- nor3(Y, A, B, C)
- nand4(Y, A, B, C, D)

Tie High, Tie Low, Row Crosser



These three cells contain no transistors. They exist merely to facilitate the operation of an automated standard cell place and route tool.

- Tie High & Tie Low

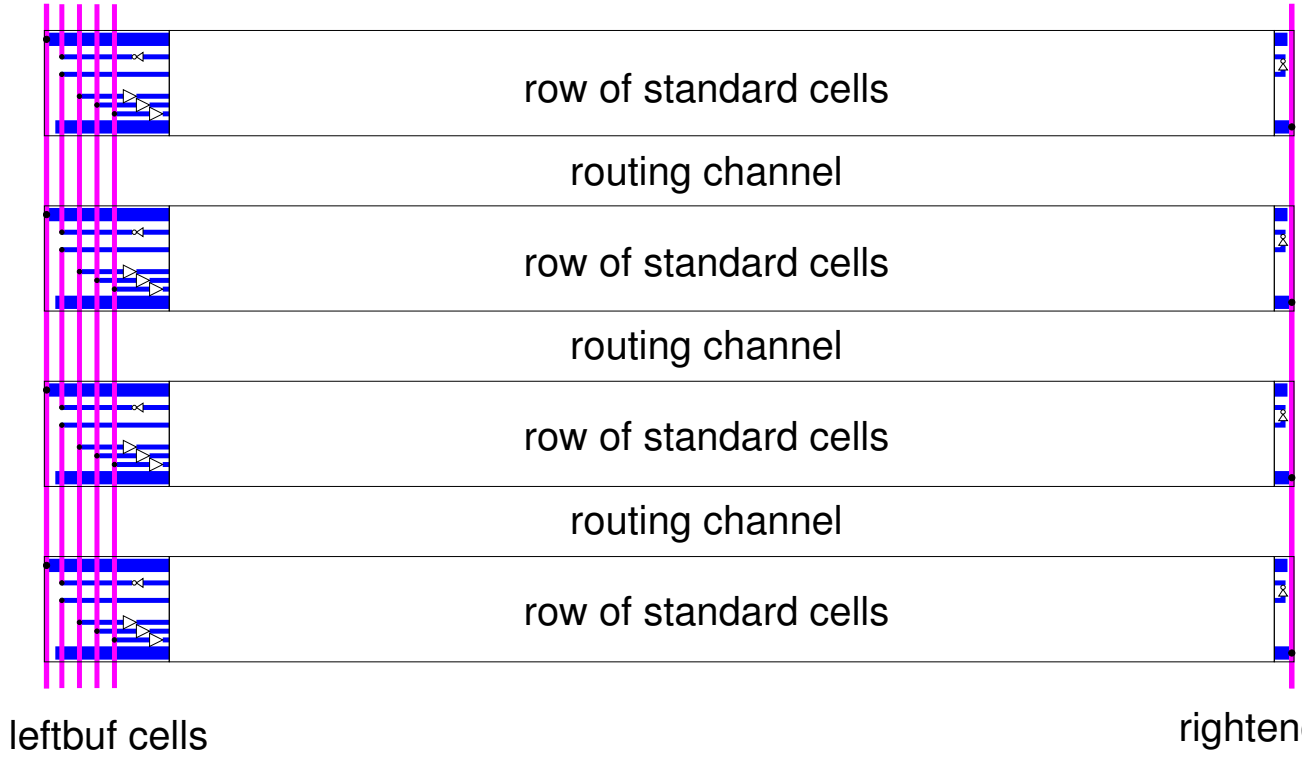
  These cells, named `tiehigh` and `tielow`, are used to tie unused cell inputs to either Vdd! or GND! as appropriate.

- Row Crosser

  This cell, named `rowcrosser`, is used where a signal needs to be routed over a row of cells.

End of Row Cells

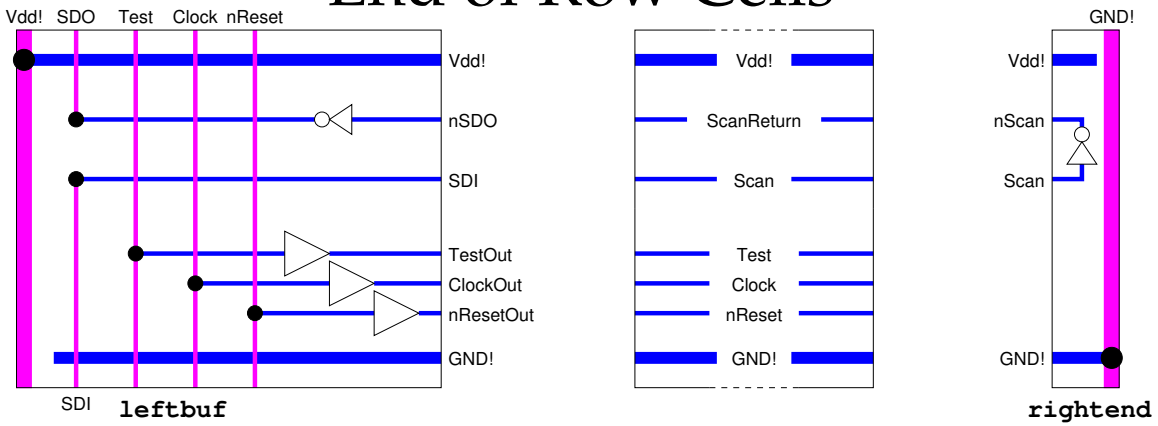
Each row of standard cells will have a `leftbuf` cell at one end and a `rightend` cell at the other.



These cells facilitate distribution of global signals: Vdd!, GND!, Clock, nReset and Test. They also include support for the routing of the scan path.

Cell Library Design

End of Row Cells



- Global Signal Distribution

  The `leftbuf` cell buffers and distributes Clock, nReset and Test.

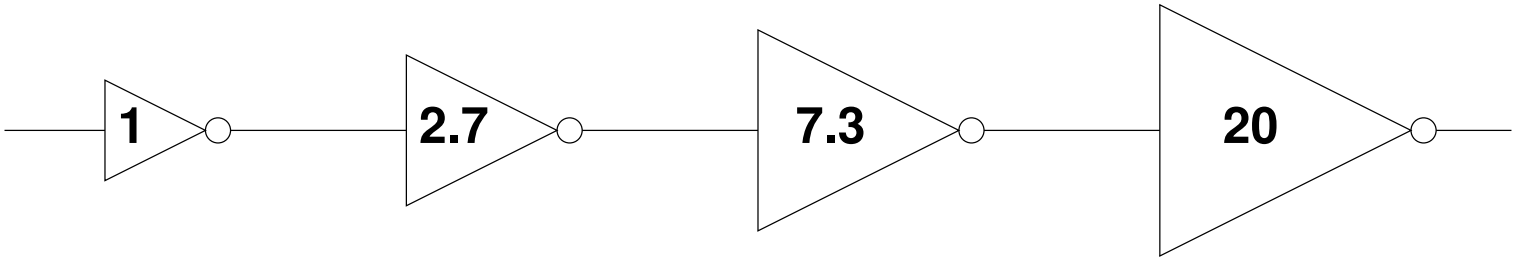
  The `leftbuf` cell distributes Vdd! while the `rightend` cell distributes GND!.

  Note: Vertical power rails in these cells should be $10\mu m$ wide to take account of the large supply currents.

- Scan Path Support.

  The inverters in the `leftbuf` and `rightend` cells provide buffering in the scan path and avoid name conflicts when the design is extracted.

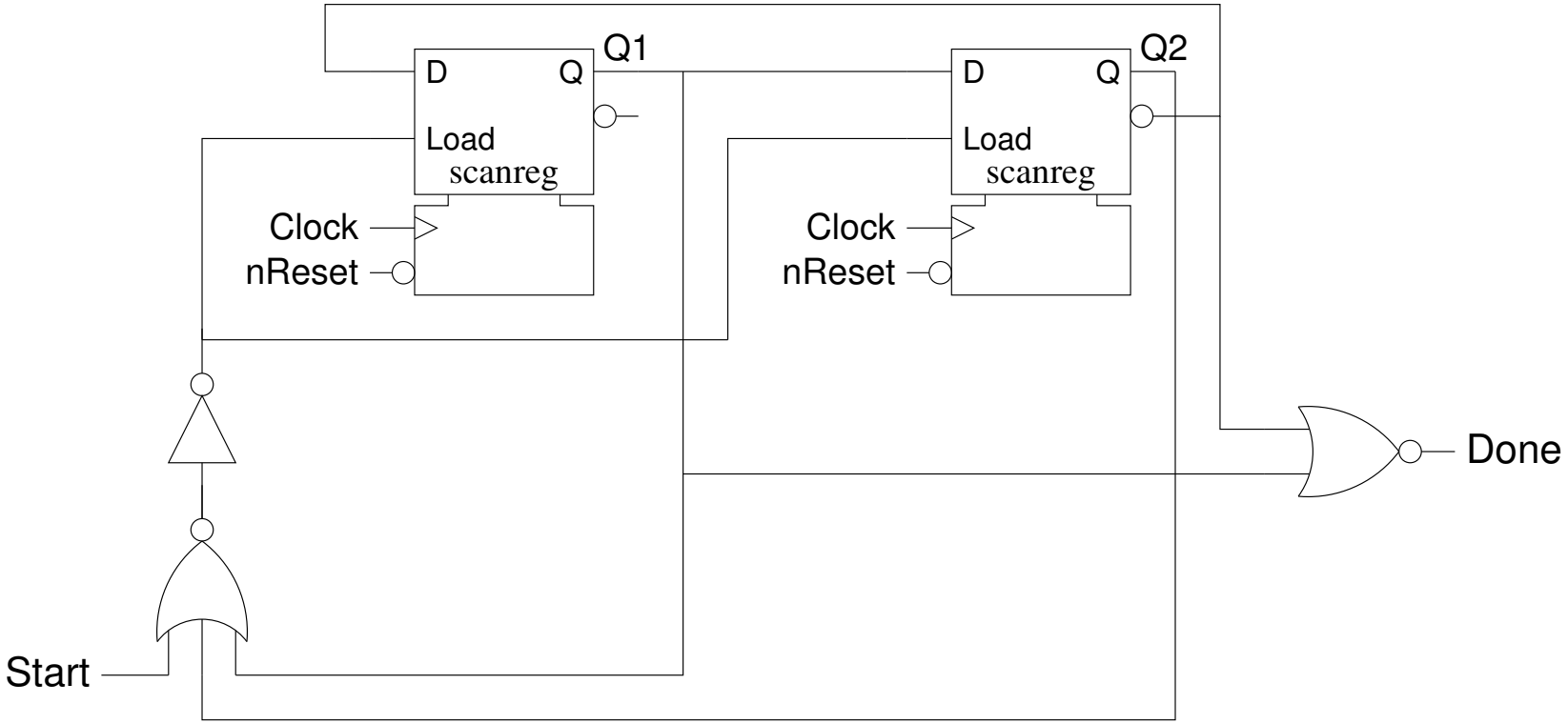
Left End of Row Cell: Buffer Design



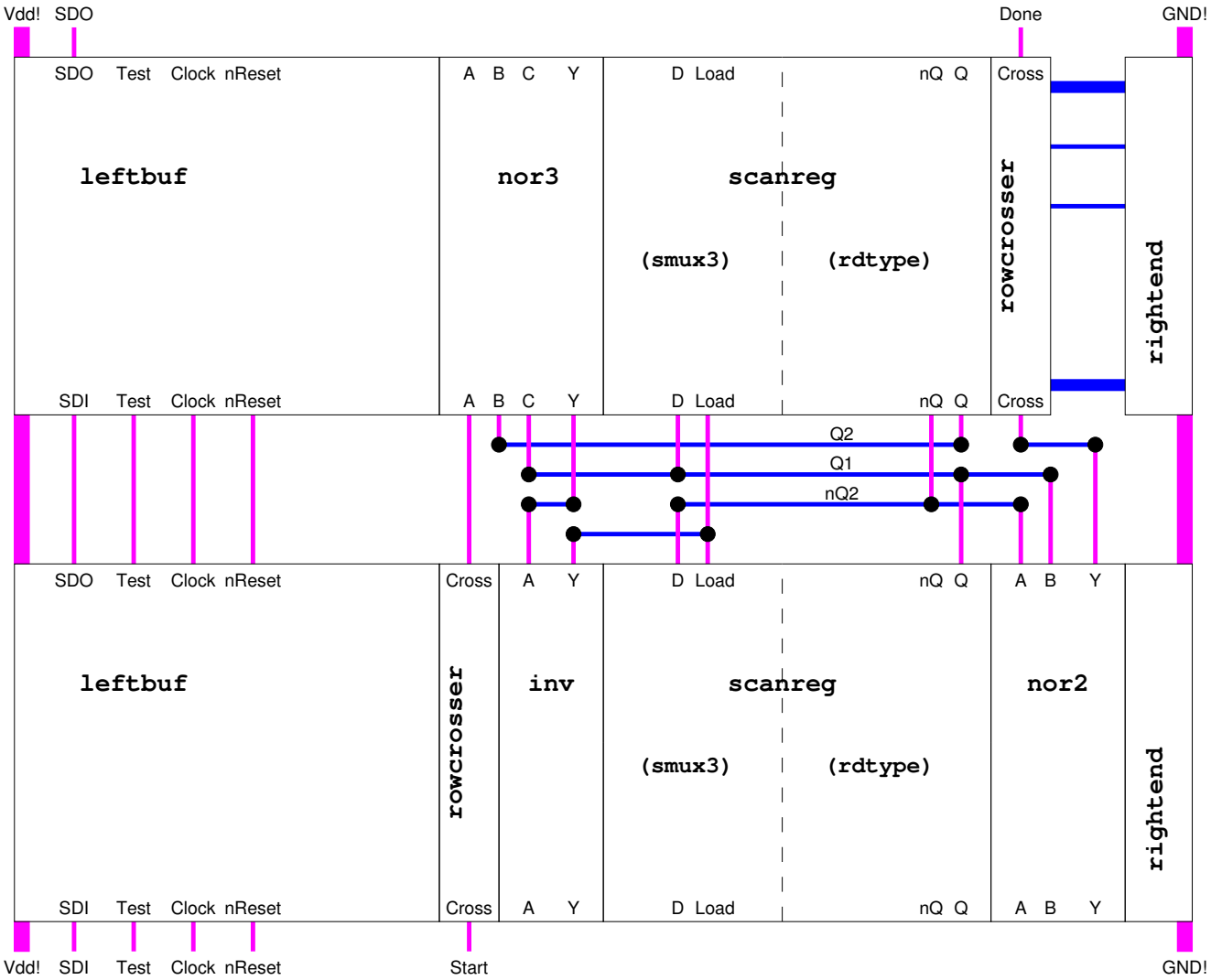
- Each non-inverting buffer is constructed from 4 inverters
- Each inverter is 2.7 times the size of the previous one
- The final inverter will drive up to 20 D-types with minimal clock skew
- The first inverter is a modified version of the standard inverter with transistor widths adjusted to give approximately equal rise and fall times (W_P = 2.9μm , W_N = 1.0μm)
- The complete leftbuf cell is designed without hierarchy

Example Design - Schematic

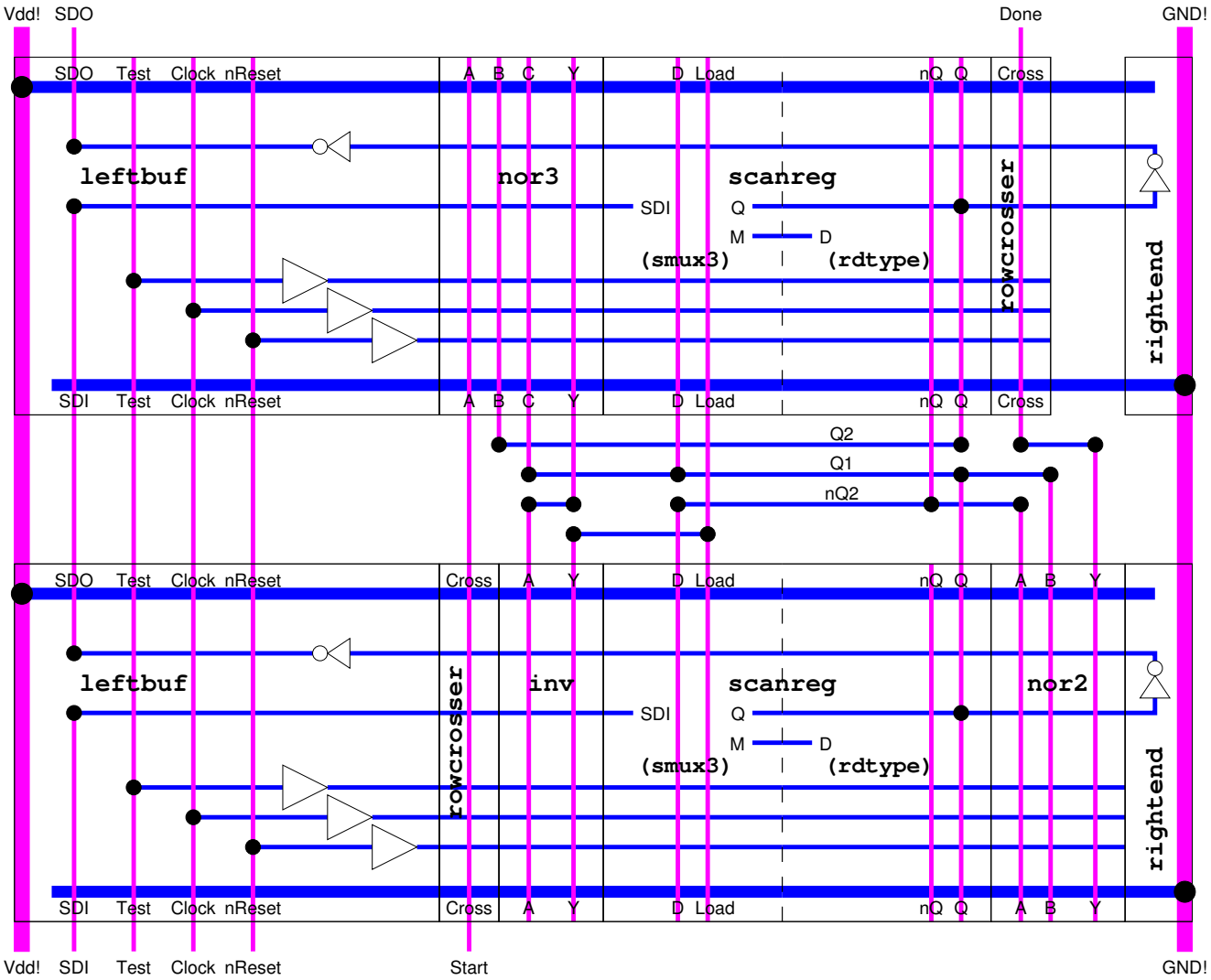
The use of the end of row cells and the row crosser cell can be seen in the following example:



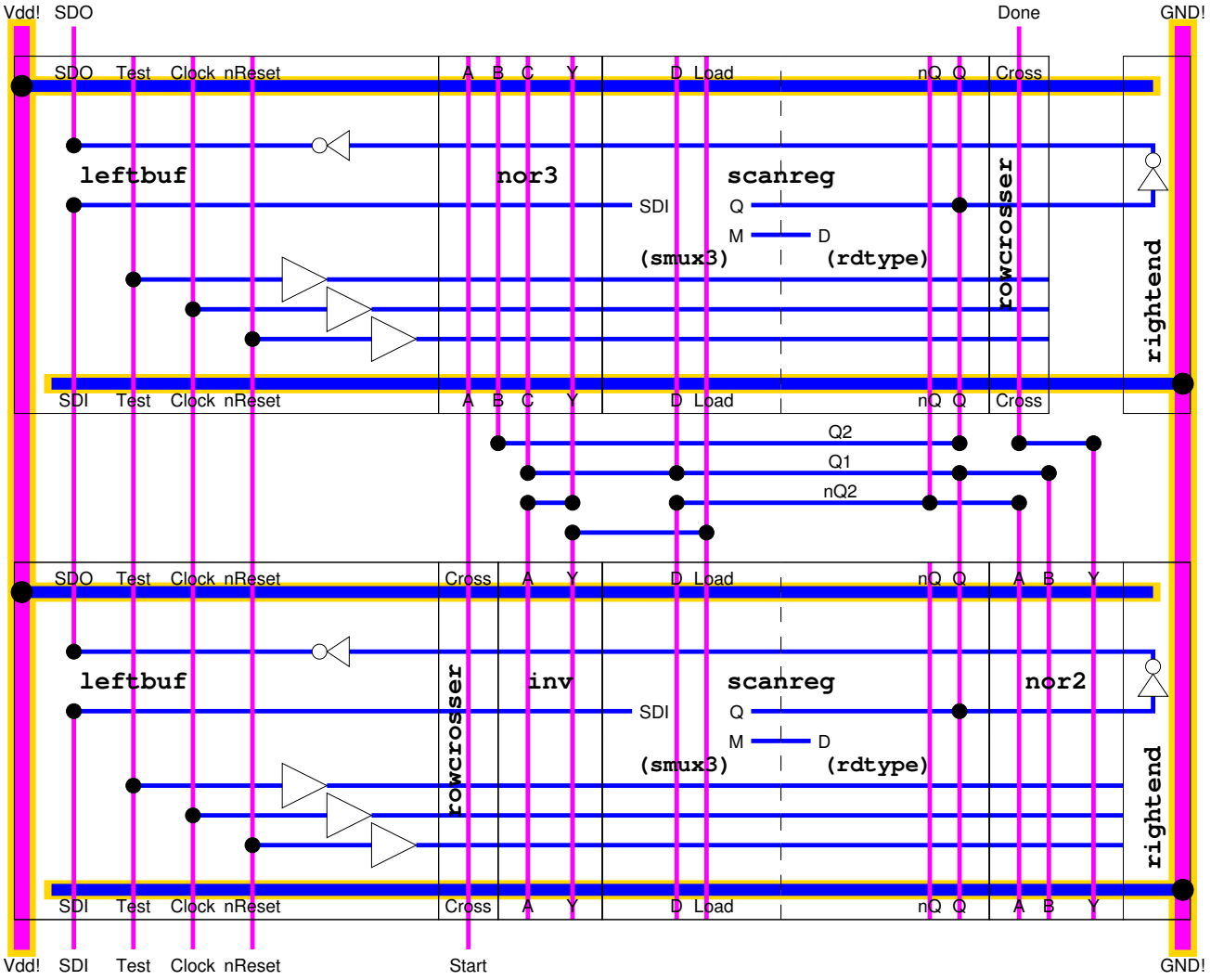
Example Design - Layout



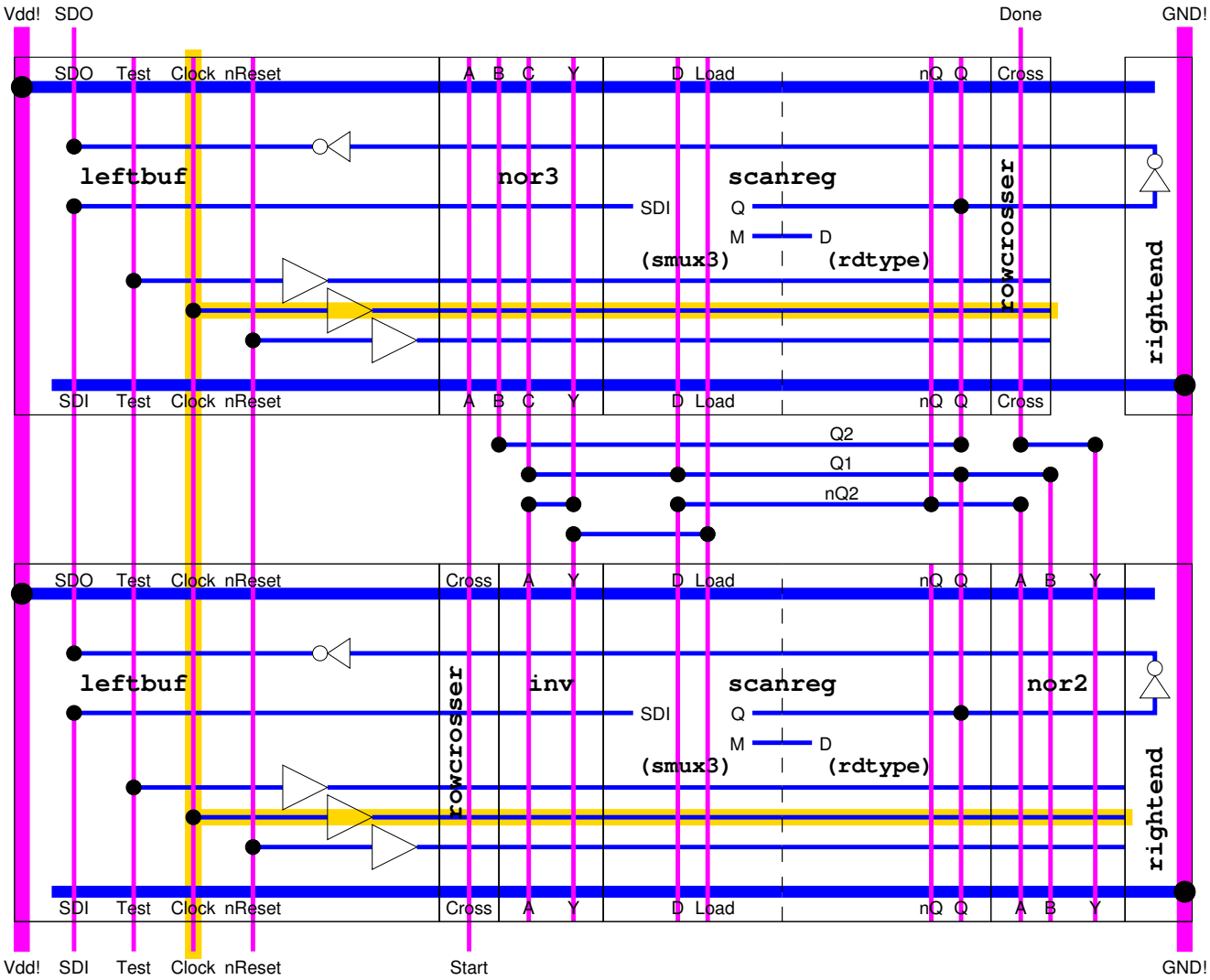
Example Design



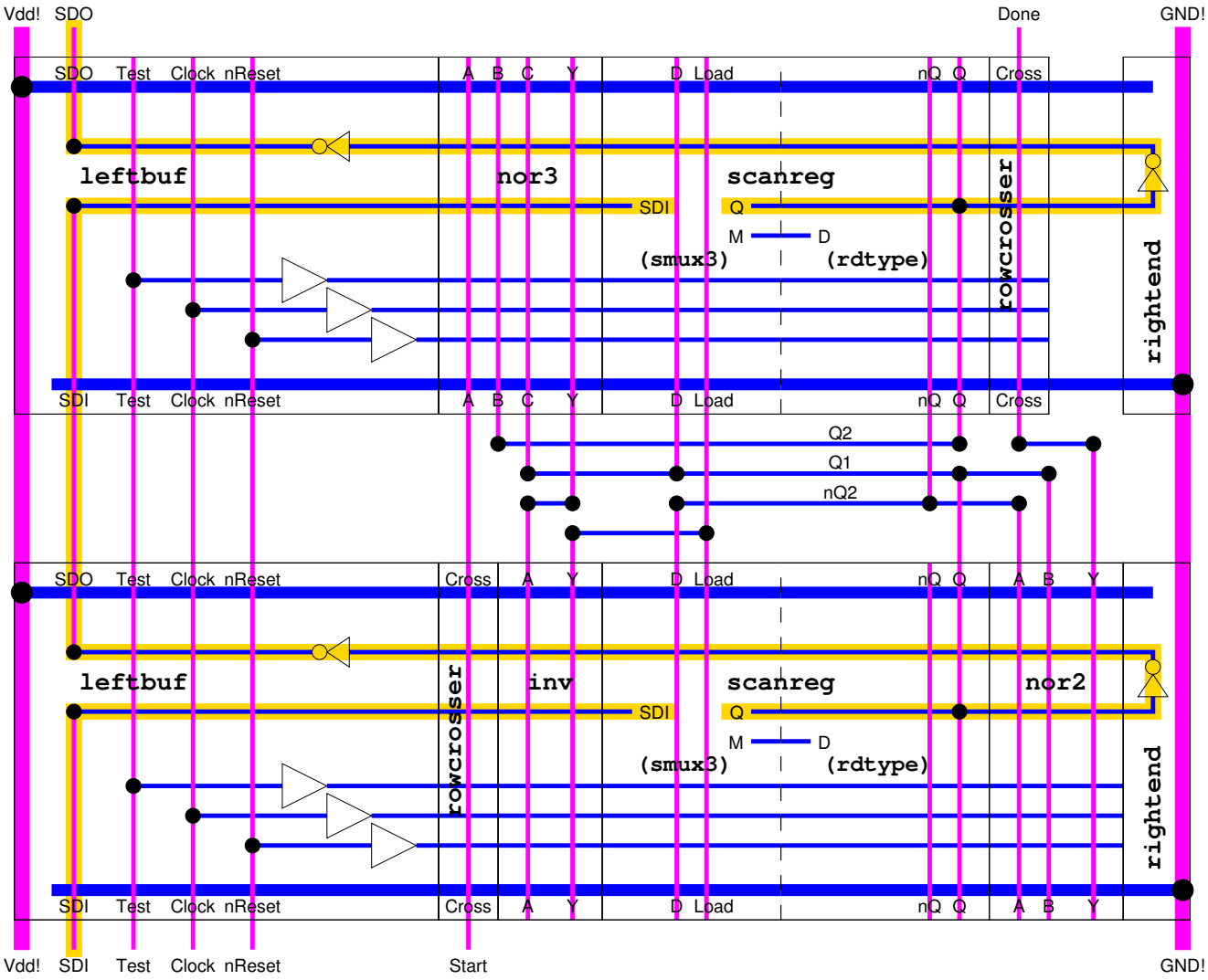
Example Design - Power Distibution



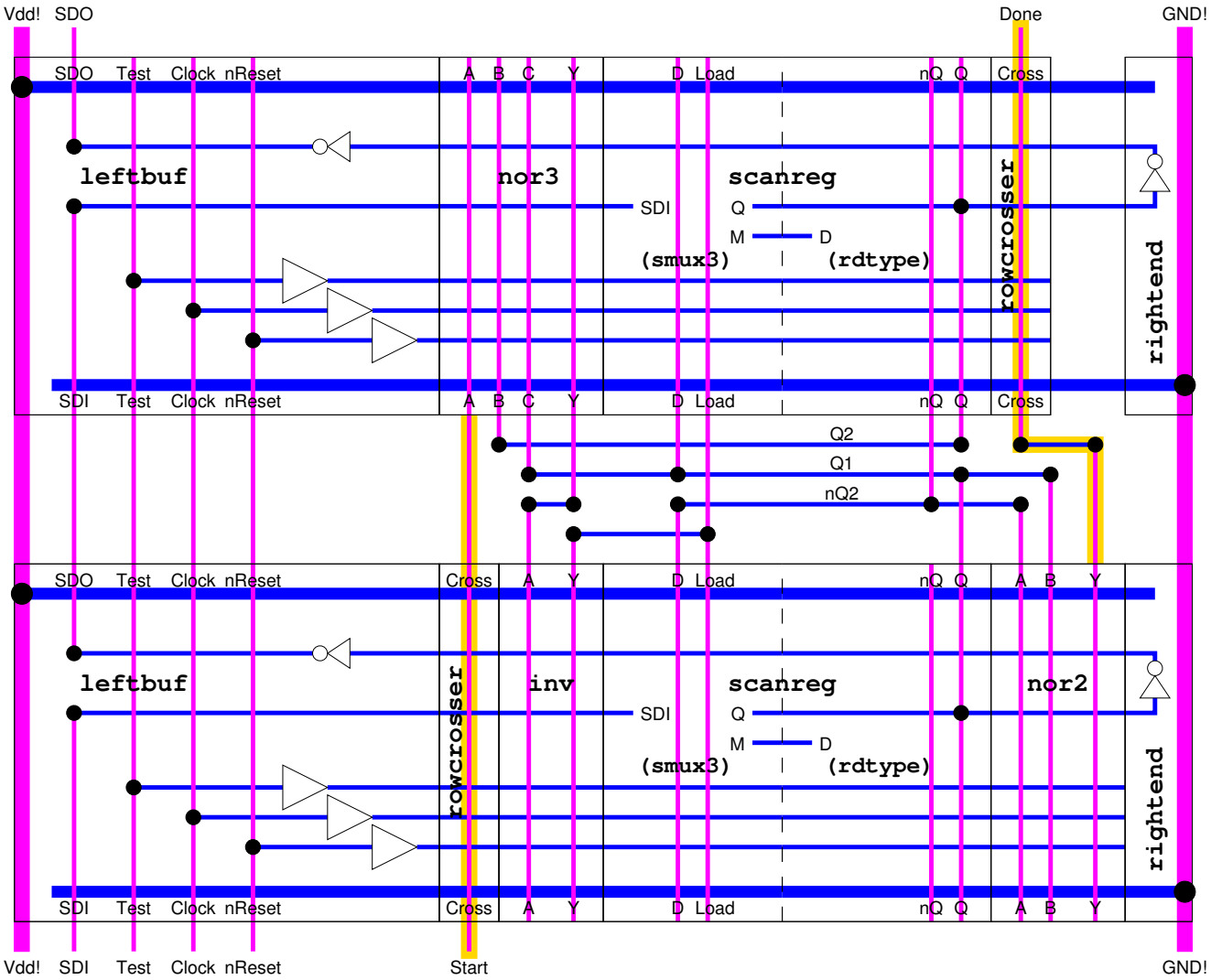
Example Design - Clock Distibution



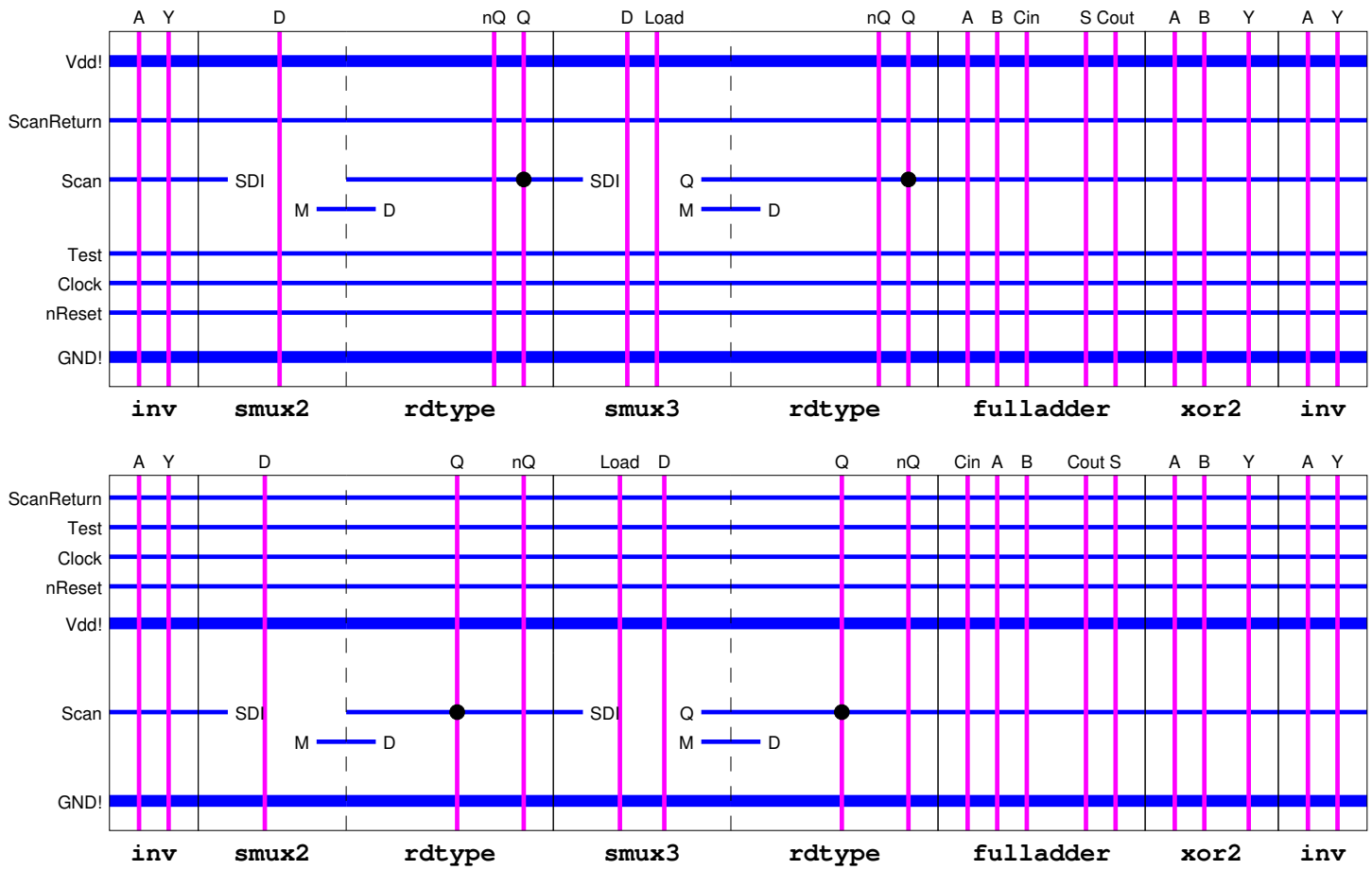
Example Design - Scan Path



Example Design - Row Crosser

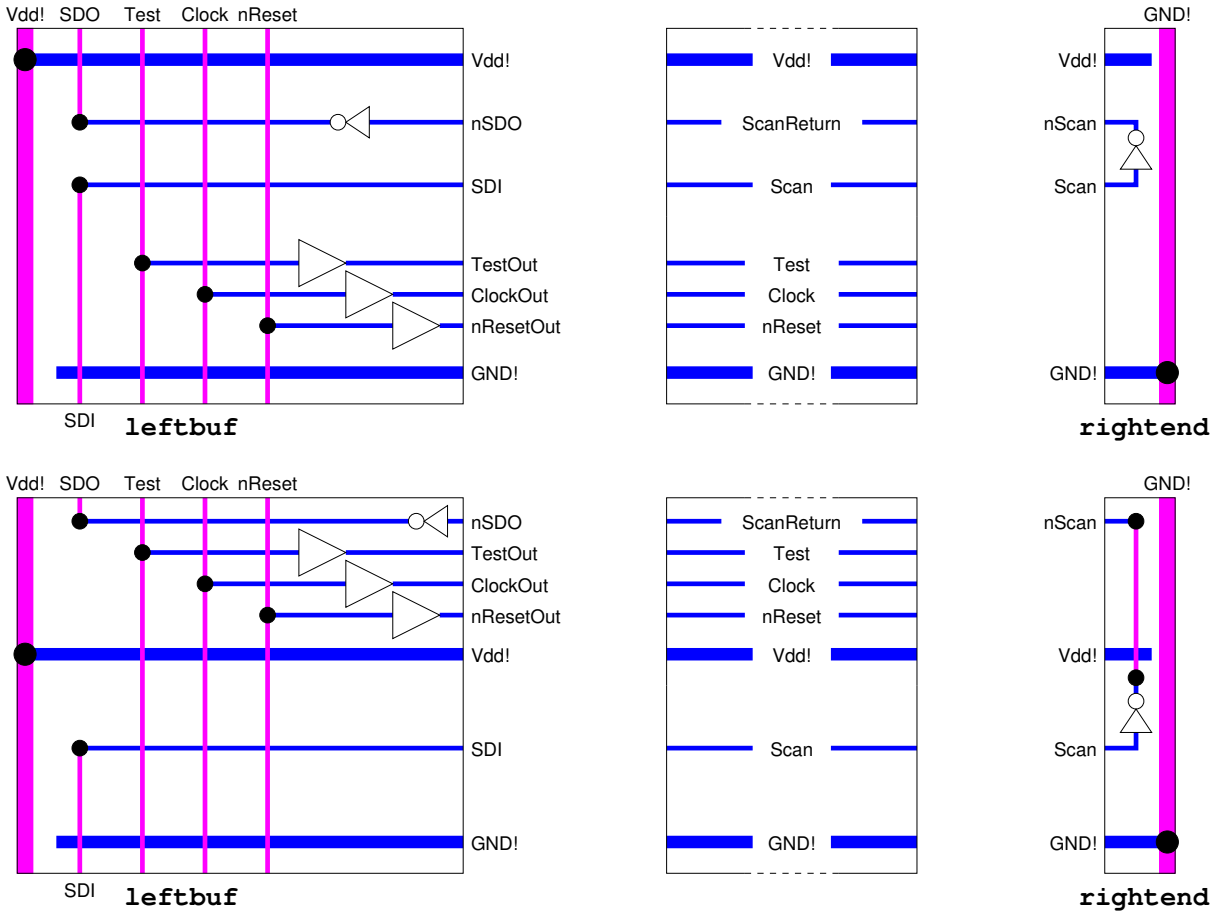


Cell Library Design



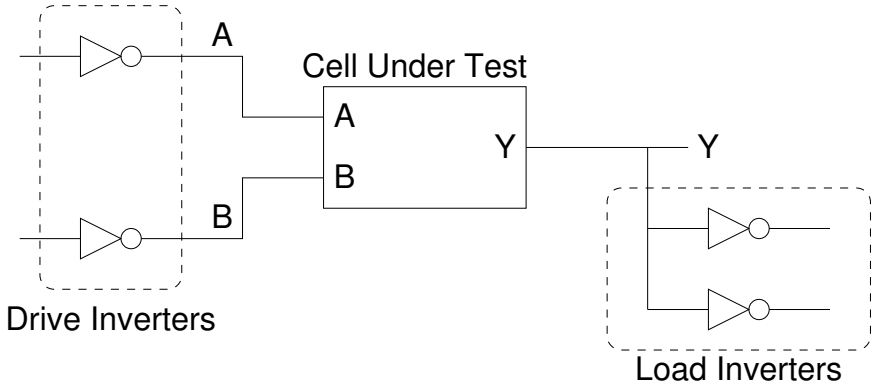
The arrangement of i/o is up to you. You must ensure the correct alignment of the horizontal i/o (especially the scan path) between cells.

Cell Library Design



The arrangement of horizontal i/o in end of row cells should match your other cells.

Characterization



Each cell should be characterized under realistic conditions. In this case your own inverter cell will be used as a standard drive/load.
When doing Spice simulations of your gates each input should be driven by an inverter[1] and each output should be loaded with 2 inverters.

For each gate output you should extract typical propagation delay, t_p, based on this drive and load.[2]

[1] inputs to these drive inverters should be relatively fast
[2] for the D-type, the propagation delay is measured from the Clock input to the Q/nQ output

Division of Labour - Teams of 5

For teams of five, the tasks will be divided as follows:

- Raw D-Type `(rdtype)`
- Two Scan Multiplexer `(smux2, smux3)` and Standard Multiplexer `(mux2)`
- Full Adder `(fulladder)`
- Half Adder `(halfadder)` and XOR `(xor2)`
- End of Row Cells `(leftbuf, rightend)`

Note that for a complete Scannable D-Type we need the raw D-Type cell from one designer and a 2 input scan multiplexer from a second designer.

It is up to you who designs each of the other cells `(inv, buffer, nand2, nor2, and2, or2, nand3, nor3, nand4, trisbuf, tiehigh, tielow, rowcrosser)`. You should use this flexibility as a means of balancing the load.

Division of Labour - Teams of 4

For teams of four, the tasks will be divided as follows:

- Raw D-Type `(rdtype)`
- Two Scan Multiplexer `(smux2, smux3)` and Standard Multiplexer `(mux2)`
- Full Adder `(fulladder)`
- End of Row Cells `(leftbuf, rightend)`

Note that for a complete Scannable D-Type we need the raw D-Type cell from one designer and a 2 input scan multiplexer from a second designer.

It is up to you who designs each of the other cells `(inv, buffer, nand2, nor2, and2, or2, nand3, nor3, nand4, trisbuf, tiehigh, tielow, rowcrosser, halfadder, xor2)`[3]. You should use this flexibility as a means of balancing the load.

[3] the `halfadder`, `or2`, `nor3` gates are optional for teams of 4

Submission – Files

- single team submission
- magic files in directory `~/design/magic/c035u/cell_lib`
  - leaf cells:

    `rdtype smux2 smux3 fulladder` *`halfadder`* `xor2 mux2 inv buffer nand2 nor2 and2` *`or2`* `nand3` *`nor3`* `nand4 trisbuf tiehigh tielow rowcrosser leftbuf rightend scandtype scanreg`
  - parent cell:

    `all`

    instances all leaf cells butted together in one row arranged as follows:

    `leftbuf inv buffer smux2 rdtype nand2 scandtype nand3 smux3 rdtype nand4 scanreg fulladder xor2 mux2 nor2 and2 trisbuf tiehigh tielow rowcrosser` *`halfadder or2 nor3`* `rightend`
- preparation script creates `handin.tar` file for web based handin

```
prepare_dicd desex3
```

Deadline for file submission is 19:00 on Wednesday 4th December.

Cell Library Databook

a single team databook with tightly controlled structure

Main Body

- Title Page
- Contents - including list of all cells with page numbers
- Introduction (max 3 sides)
- Datasheet (1 per cell - 1 side each)
- Summary of results (max 1 side)

Appendix A

- Team Management (max 1 side)
- Division of Labour Form (1 side)

Appendix B

- Design detail for major cells (max 2 sides per major cell)

Cell Library Databook

- Title Page

  Cell Library Databook

  by Team C1

  A. Smith (ac6j10)
  B.C. Legrand (bcl12i10)
  Zhang X.Y. (xyz11k10)

- Introduction (max 3 sides)

  Introduces the library and the databook.

  Gives detail of features common to all cells in the library (e.g. cell height)

  Picture showing position of signals and other cell structures such as the N-well as they exist at the interface between cells

Cell Library Databook

- Datasheet (1 side per cell[4])

  Include datasheets for cells to be used by the ASIC designer (e.g. scanreg) rather than cells only used by library cell designers (e.g. rdtype, smux3).

  - cell name
  - cell description
  - cell designer(s)
  - symbol[5]
  - gate level circuit diagram[6]
  - abstract

    black box view showing dimensions
  - table of measured parameters
  - verilog simulation waveform

[4] trivial support cells may be combined on one page

[5] no symbol exists for leftbuf, rightend, tiehigh, tielow, rowcrosser

[6] no gate level circuit diagram required for nand, nor, inv, tiehigh, tielow, rowcrosser

Cell Library Databook

- Design detail for major cells (max 2 sides per major cell)

  `rdtype, smux2, smux3, fulladder, halfadder, xor2, leftbuf`

  A sub-section should exist for each designer, dealing with the major cells they have designed.

  For each cell the following information should be included in the design detail:[7]

  - transistor level circuit diagram
    with comments on derivation if appropriate
  - stick diagram
  - details of circuit for loaded simulation
  - results of loaded simulation
    numerical and graphical

[7] take care with diagrams, don't scan/capture my schematics, avoid screen capture of simulation and layout if possible, and avoid using inappropriate pre-defined circuit symbols (e.g. MOS transistors with visible substrate connection).

Cell Library Databook

- Presentation
  - Use full sentences and paragraphs
    note form is acceptable on the datasheets but not in design detail sections or elsewhere in the databook
  - Cover
    The handin system allows you to print a standard cover for this report
  - Binding
    This report must be bound. If printing double sided, then a neat line of staples down the long edge of the report will be sufficient.

Documentation for this exercise should be submitted to the school office.

Deadline for documentation is 16:00 on Tuesday 10th December.

Final note: extra credit may be given for advanced characterization e.g. Δt_p *,* t_{setup} *(for sequential gates), or clock skew simulations (for left end of row cell)*