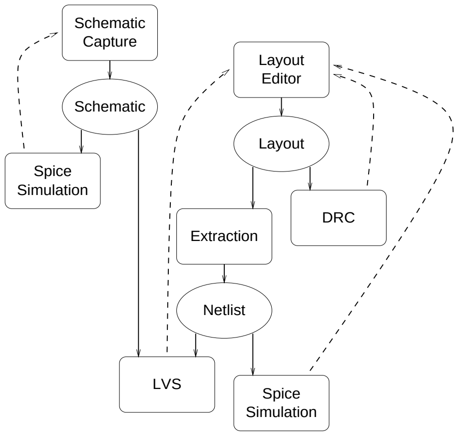
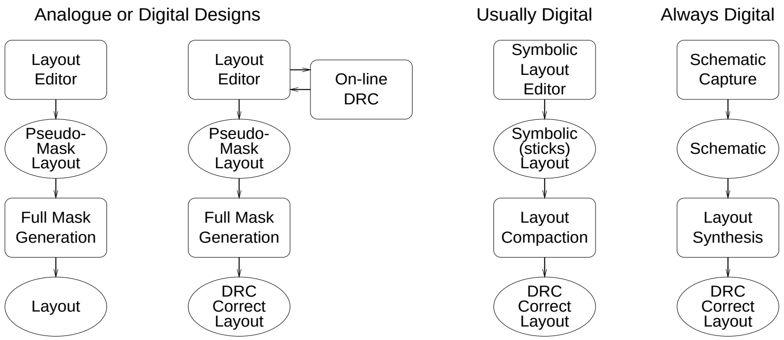
Design Flow – Custom Block Design



- Verification is the key to producing *right first time* silicon.

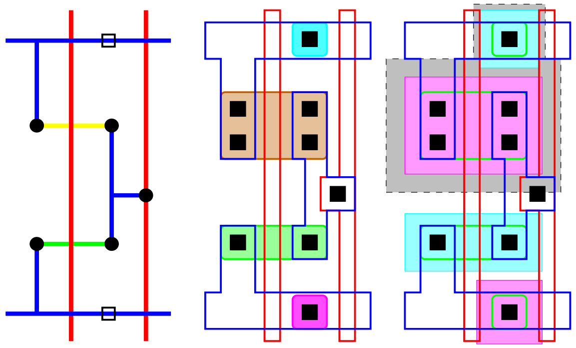
Design Flow – Custom Block Design

Design Automation Tools



- Automation tools can help to improve productivity for custom designs.

Pseudo-Mask Layout



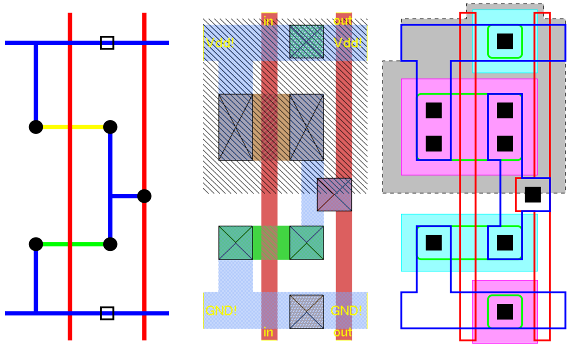
- Pseudo Masks:

  N Diffusion P Diffusion N Ohmic P Ohmic

- Auto Generated Masks:

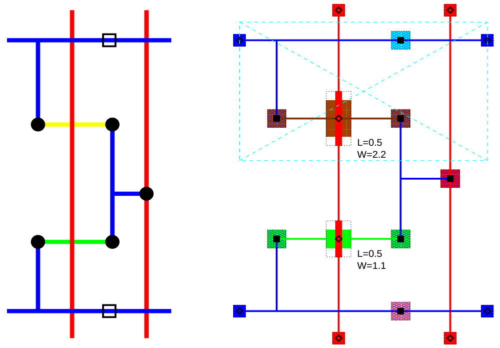
  Active Area N Select P Select N Well

Pseudo-Mask Layout with On-line DRC (Magic)



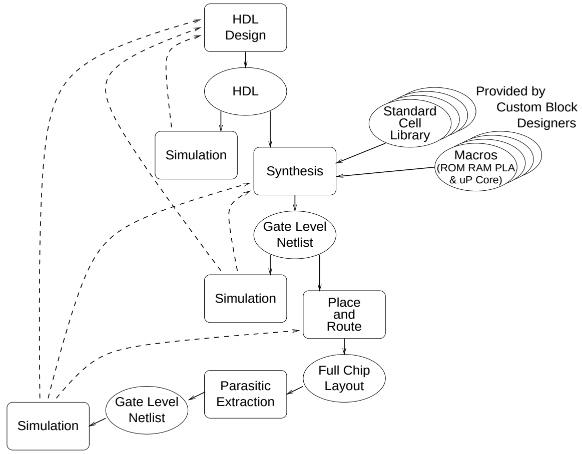
- Log style design (sticks with width).
- Each contact *tile* creates a three level structure consisting of the conductors to be connected and the appropriate cut (or cuts) in the intervening insulation.
- DRC errors are flagged immediately - dramatically reduces design times.

Sticks Layout – Symbolic Capture



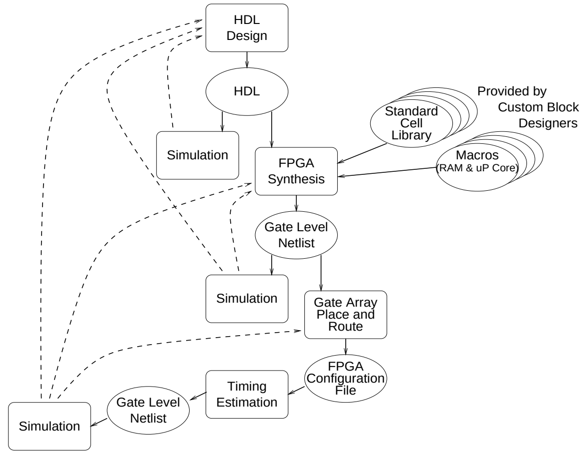
- Transistors are placed and explicitly sized.
  - components are joined with zero width wires.
  - contacts are automatically selected as required.
- A semi-automatic compaction process will create DRC correct layout.

Design Flow – Semi-Custom Design



- Extensive use of automation with manual intervention.

Design Flow – FPGA Design



- Almost identical flow for FPGAs.

Hardware Description Language

Design of large semi-custom digital systems usually uses HDL rather than schematics.

- **Register Transfer Language**

  A multiplication algorithm is expressed in generic RTL. In RTL we describe what will happen on the next active clock edge:

  $A \Leftarrow A << 1$

  $B \Leftarrow B >> 1$

  if $(B[0] = 0)$ then $P \Leftarrow P + A$

  We can convert our generic RTL to a real HDL such as VHDL or Verilog.

- **Verilog**

```
always @(posedge CLOCK)
  begin
    A <= A << 1;
    B <= B >> 1;
    P <= (B[0]) ? P+A : P;
  end
```

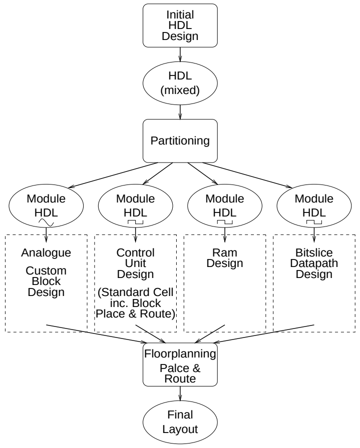
Hardware Description Language (Verilog)

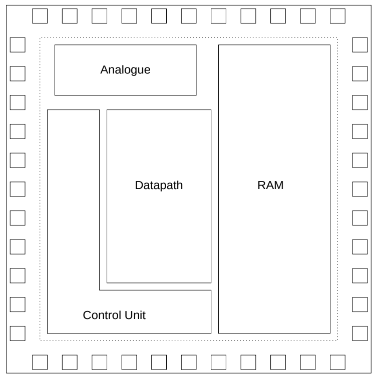
```
// Verilog behavioural model of a 4 bit multiplier

module multiply (P, READY, ACK, Ain, Bin, STROBE, CLOCK, nRESET);

output [7:0] P;
output READY, ACK;
input [3:0] Ain, Bin;
input STROBE, CLOCK, nRESET;

reg [7:0] P;
reg [6:0] A;
reg [3:0] B;
reg ACK, READY;
wire START, LAST;

assign START = STROBE && (B[3:1]==0);
assign LAST  = (B[0] || ACK) && (B[3:1]==0);
```



```
always @(posedge CLOCK or negedge nRESET)
  if (!nRESET)
    begin
      A <= 0; B <= 0; P <= 0; ACK <= 0; READY <= 0;
    end
  else
    begin

      ACK   <= START;
      READY <= LAST;

      A <= (START) ? Ain : A << 1;
      B <= (START) ? Bin : B >> 1;

      if (ACK)
        P <= (B[0]) ?   A : 0;
      else
        P <= (B[0]) ? P+A : P;

    end
endmodule
```


Design Flow – Large Custom Designs





- Divide and Conquer.