



Research Consortium in Speckled Computing

Transceiver ASIC Design for the 5Cube

Matthew Barnes and Martin Ling

University of Edinburgh

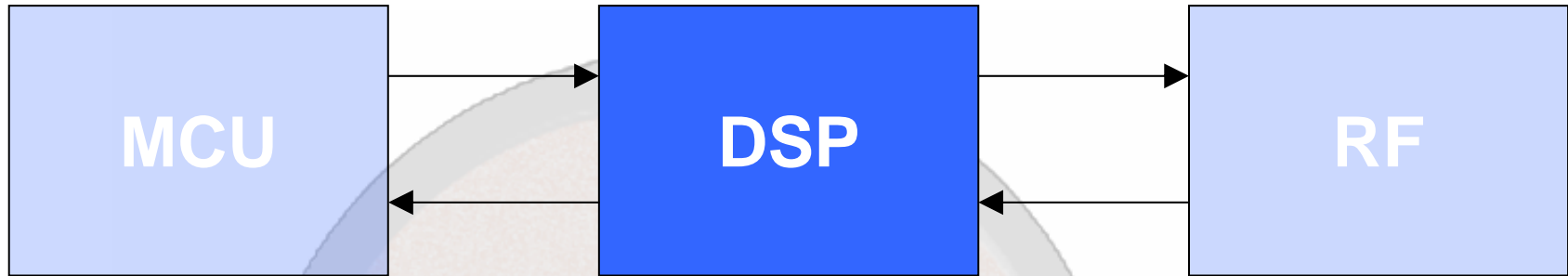
m.barnes01@gmail.com / m.j.ling@ed.ac.uk



Overview

- Introduction to the process of engineering a digital ASIC design.
- High level specification & interfaces
- Division into subcomponents
- HDL behavioural descriptions
- Synthesis to standard cells
- Placement and routing
- Test procedures at each stage
- Limitations and future work

High level specification



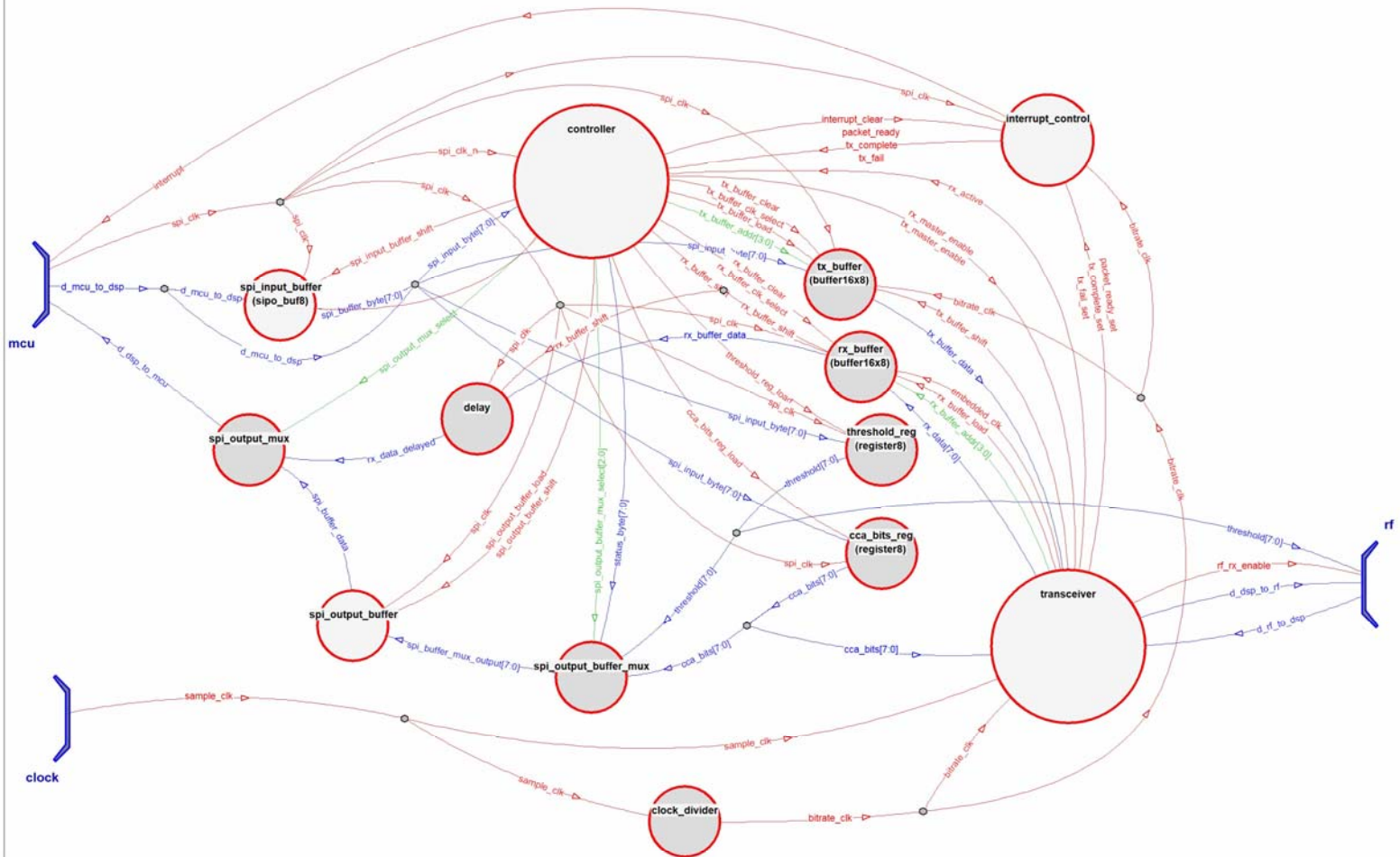
- Functions to be carried out?
- Interfaces to other components?
 - Signals
 - Low level protocols
 - Command sets
- Written, human-readable specification.

Division into subcomponents (1)

- Divide functionality into small components.
 - Separate problem into more manageable tasks
 - Reuse basic components
 - Separate control sequencing from data path
 - Separate clock domains
- Design may be restructured as it evolves.
- Division of design does not affect efficiency!
- End result will be flattened after synthesis, removing overhead.

Division into subcomponents (2)

Page 1 of 1



University of Strathclyde			
Project:	Design:	FiveCube_dsp	
Designer:	Louise Crockett and Neil MacEwen:	FiveCube_dsp_TOP	Block Version:
Version:	V0.0	Notes:	
Date:			



HDL behavioural descriptions

- HDLs (Hardware Description Languages)
 - VHDL and Verilog most common.
 - Describe behaviour of components.
 - Instantiate and connect components.
- HDLs also used to write testbenches.
 - Apply stimuli to component interface.
 - Monitor outputs and check assertions.
 - Black box testing with interchangeable designs.
- Much HDL code can be automatically generated.
 - Structures of connected components.
 - Finite state machines.

VHDL example

-- 8-bit shift register

```
shift_reg_8 : process(clk, load, shift, data_in_p, data_in_s)
begin
  if reset = '1' then
    data <= (others => '0');
  else
    if rising_edge(clk) then
      if load = '1' then
        data <= data_in_p;
      elsif shift = '1' then
        for i in 7 downto 1 loop
          data(i) <= data(i - 1);
        end loop;
        data(0) <= data_in_s;
      end if;
    end if;
  end if;
end process;
```

VHDL testbench example

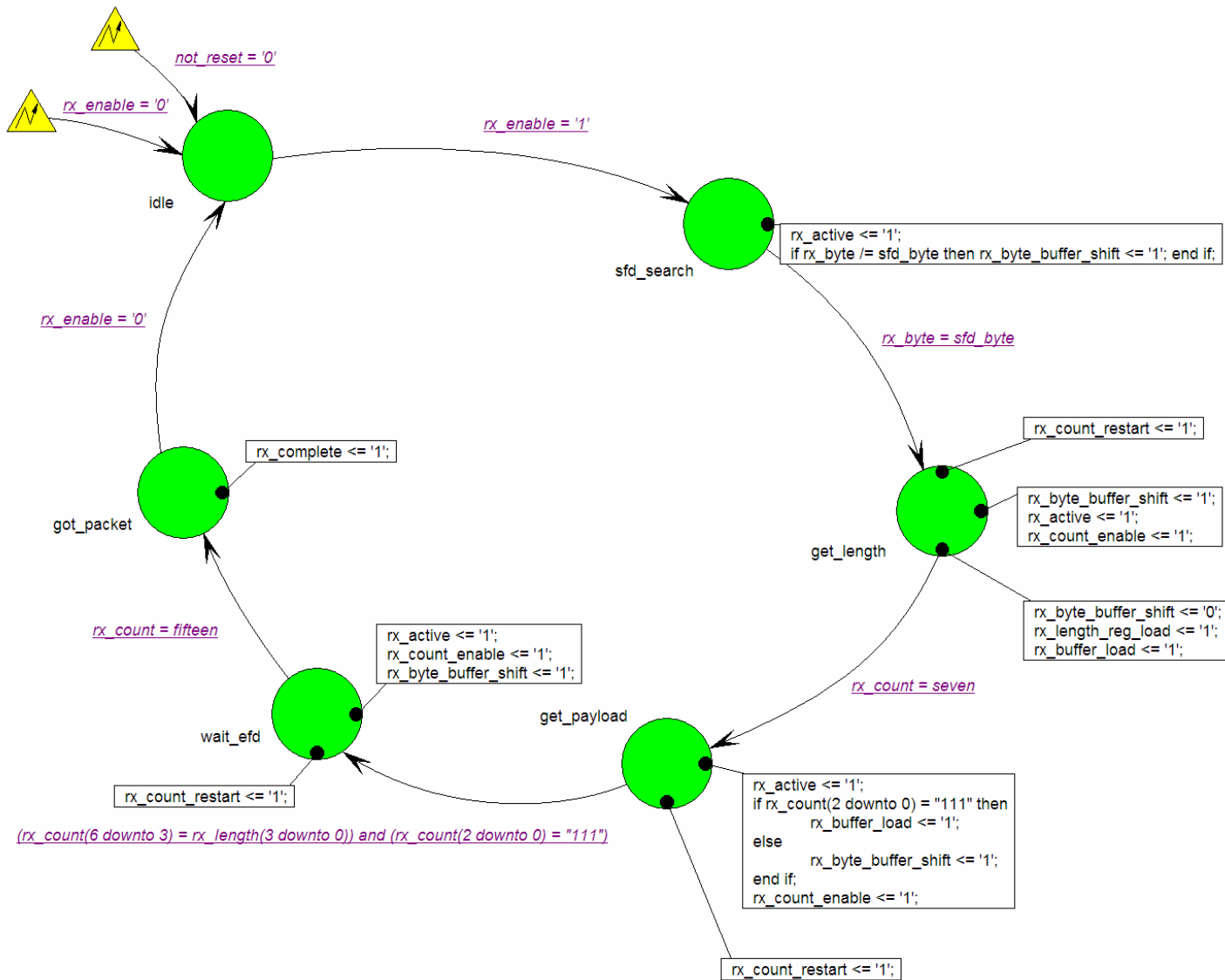
```
-- Write a byte over SPI
```

```
procedure spi_write_byte(value : in std_logic_vector (7 downto 0)) is
begin
  for i in 7 downto 0 loop
    spi_clk <= '0';
    d_mcu_to_dsp <= value(i);
    wait for SPI_CLK_PERIOD;
    spi_clk <= '1';
    wait for SPI_CLK_PERIOD;
    spi_clk <= '0';
  end loop;
end procedure;
```

```
-- Try to turn on receiver; should return success.
```

```
procedure set_rx_on is
begin
  spi_command(RX_ON, spi_rx_buffer);
  assert (spi_rx_buffer = SUCCESS)
  report "Failed to turn receiver on."
  severity ERROR;
end procedure;
```

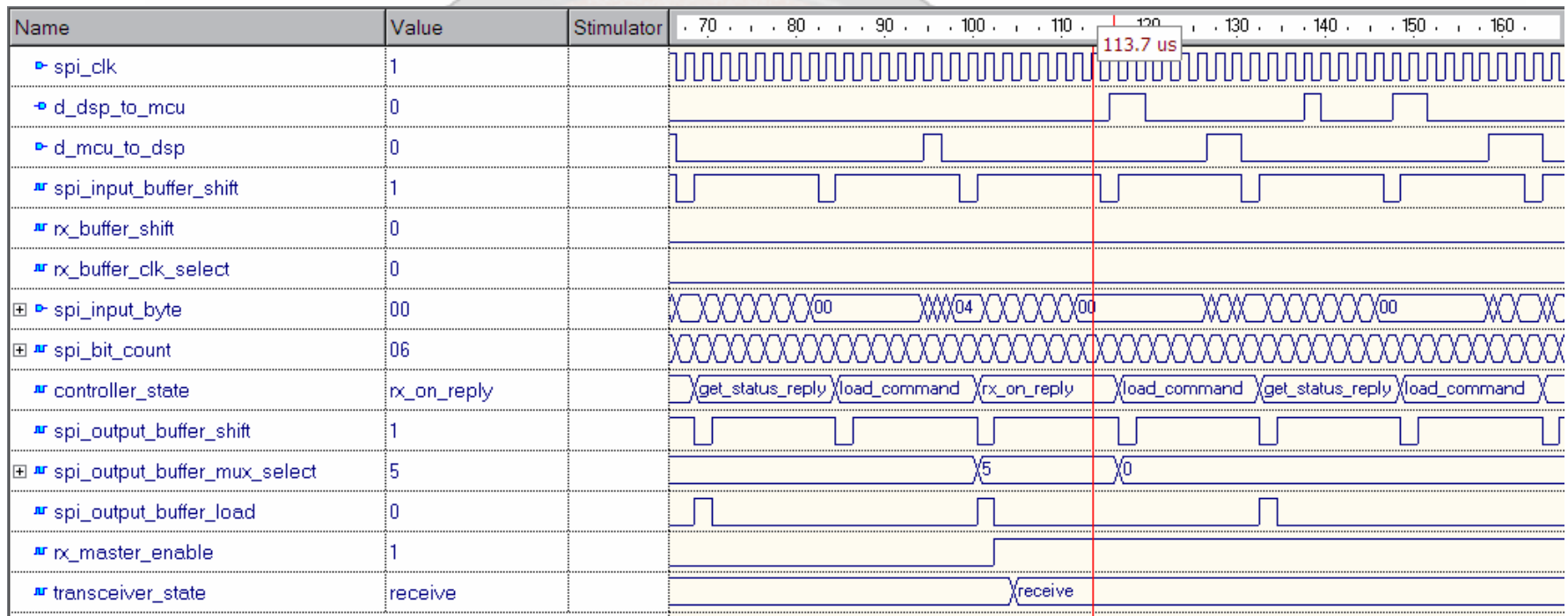
State machine example



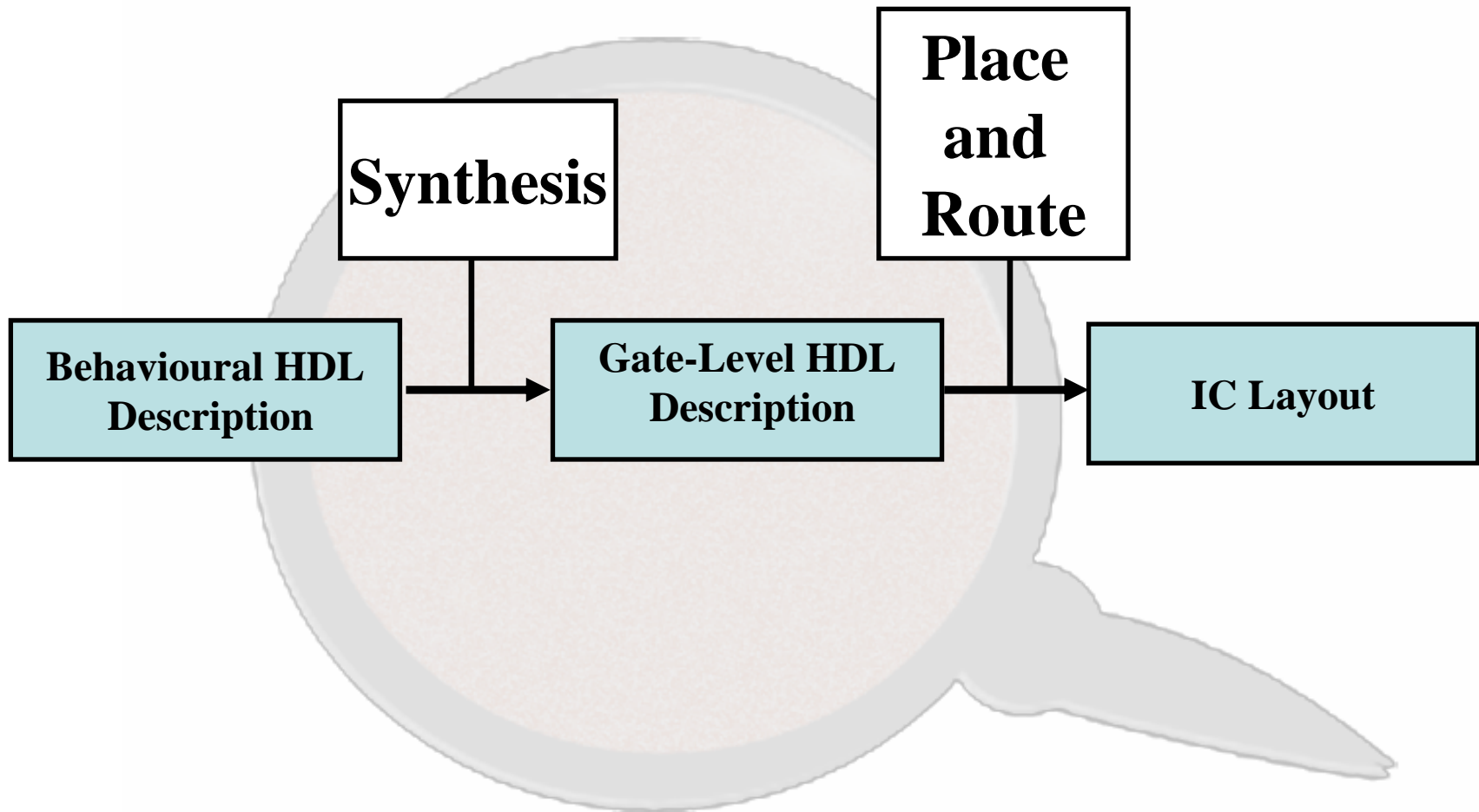
Simulation Testing

- HDL code for testbench and component compiled and simulated.
- All signals and variables can be monitored and logged.
- HDL debugging with breakpoints and stepped execution.
- Internal and testbed assertions checked during simulation.

Simulation example

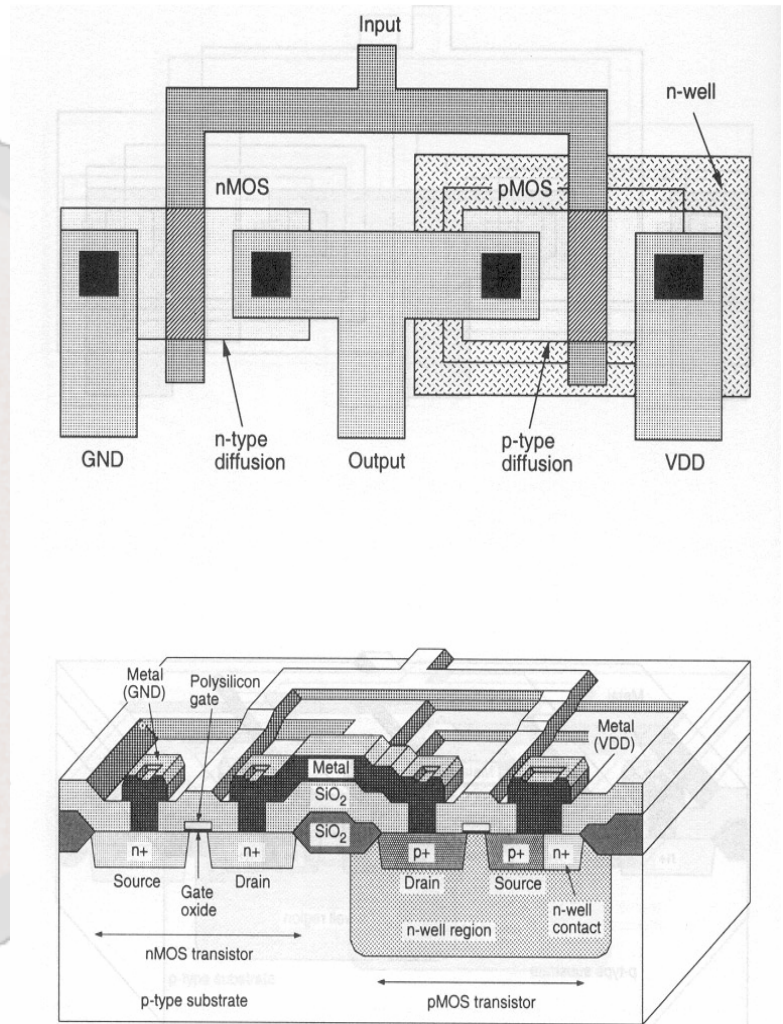


Design Flow



A Very Brief Look at CMOS ICs and MOS Transistors

- CMOS IC's are from of a number of layers of metal, insulating silicon-oxide and silicon (thus MOS)
- NMOS and PMOS Transistors are formed from doped areas of silicon, using the lowest metal layer to provide contacts (again thus MOS)
- Digital Gates are made from transistor circuits.
 - CMOS inverter example on the left.
 - The top shows a top-down view of layout of different layers
 - The bottom is an idealised view of the cross-section.



Standard Cell ASIC Design

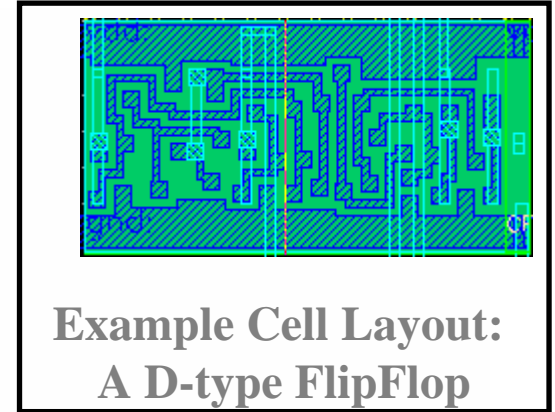
- Standard Cell Libraries

- Logic Cells

- Areas of layout with a defined logic function or purpose.
 - HDL library of behaviour and timing in standardised format.
 - Saves us all creating an AND gate...

- IO Pad Libraries

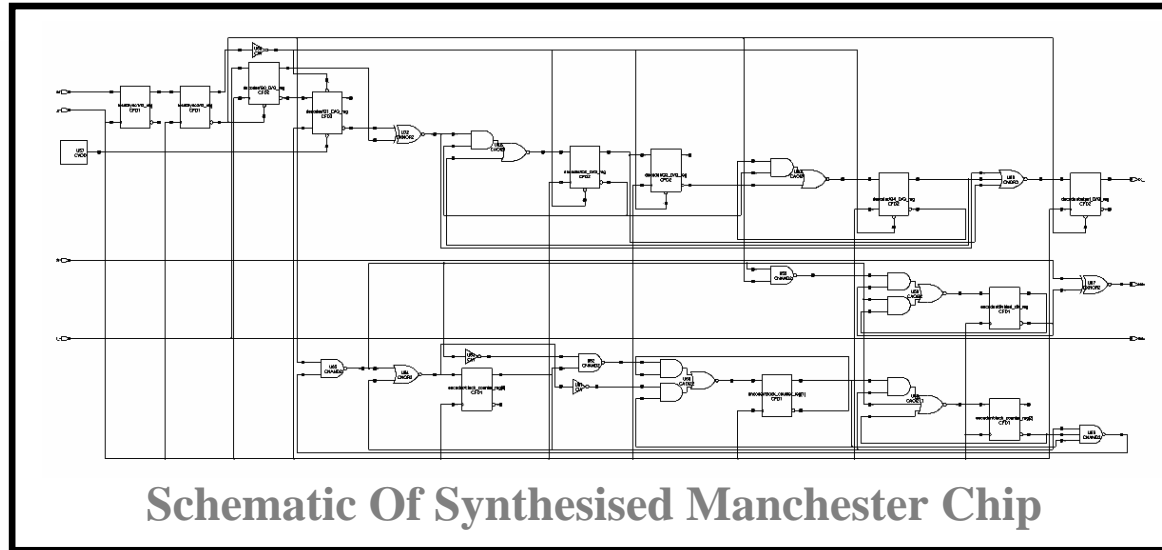
- Connection to outside world,
 - Bonding pads to connect to package or other unpackaged ICs
 - Supply and Protection Circuitry



Synthesis

- Behavioural Definition to Gate-Level Netlist
 - Synthesis is an automated process
 - Gates and Cells from library definitions
 - Net-list defines instances of cells from library and interconnecting wires
 - EDA tools are only able to synthesise a subset of behavioural HDL.
 - Design hierarchy can be removed to optimise the design

Testing the Synthesised Design

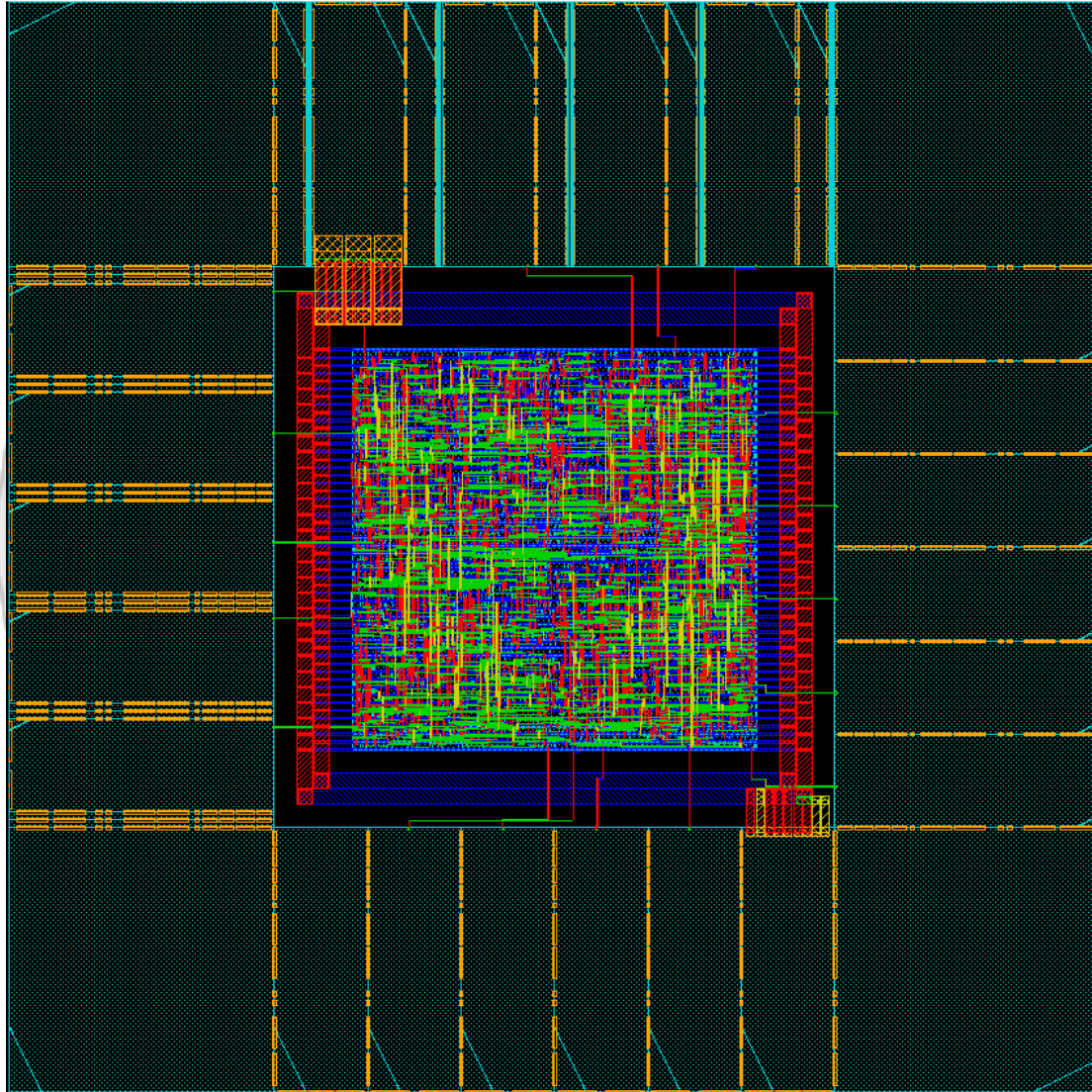


- Simulation of the gate level description
 - The behavioural description of each cell from the Cell Library is used
 - Timing Information in HDL standard
 - Black Box Testing is done using the same test-bench as the behavioural design
 - Verification of synthesised architecture

IC Layout 1: Floorplan and Partitioning

- From Gate Level HDL to IC Layout
- First Stage is to define and Partition die area
 - Logic is situated in the central ‘core’
 - Perimeter of IO pads for connection to package
 - Pad and Core Limited designs, area specification
 - Area between IO and core is left empty of cells
- Power Planning and Routing
 - Supply Rings around the Core
 - Require Sufficient Supply Pads to power core and IO cells

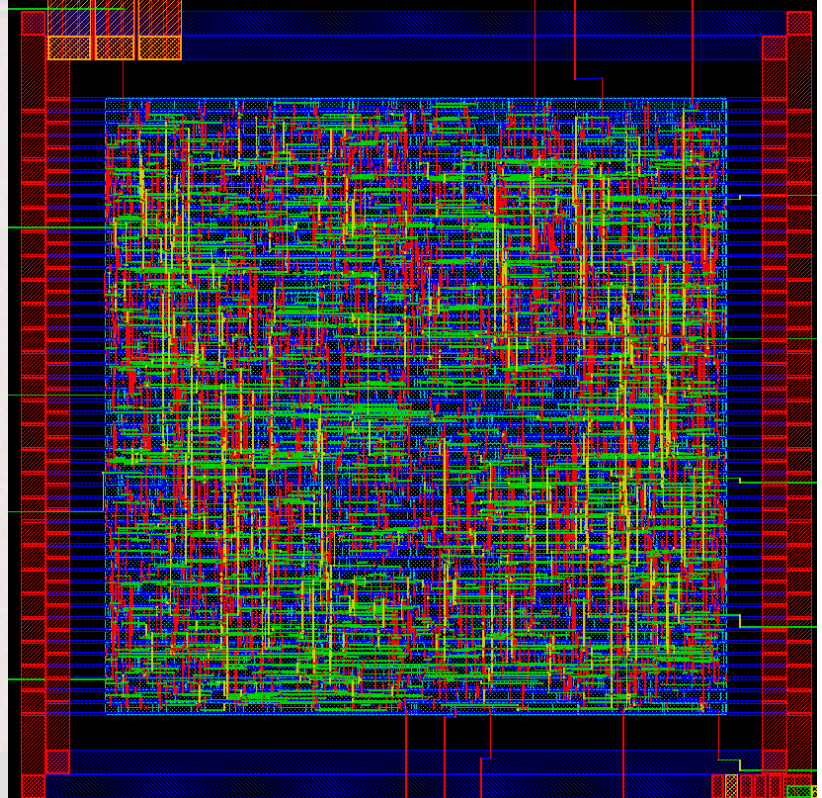
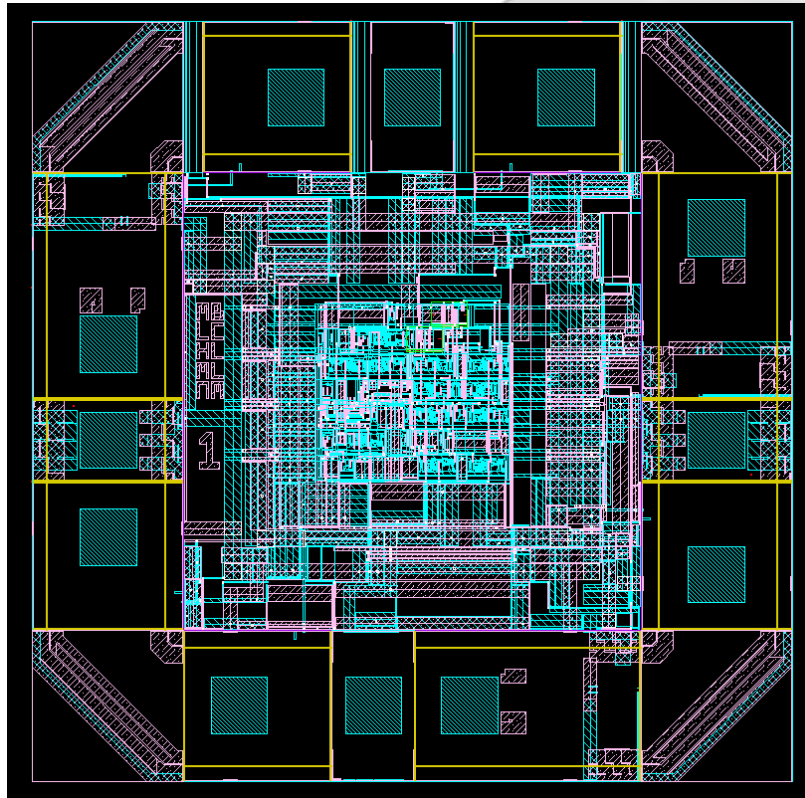
Example Layout 1: Floorplan



IC Layout 2: Cell Placement and Signal Routing

- Logic Cells Placed
 - Automated Algorithmic Process
 - Cells placed to satisfy timing constraints on special signals: clocks, resets
 - Fill empty areas of IO ring and core with filler cells
- Signals Routed on Metal Layers
 - Automated Algorithmic Process
 - Must fit all required paths around blockages of power routes and cells
 - Fill empty areas of metal with dummy filler to prevent damage during fabrication
- Iterative Process

Example Layout 2: Place and Route



Final Layout and Testing

- Gate Level HDL again extracted
 - Verify the Layout against the Schematic, LVS.
- Timing Information extracted based on defined process parameters.
 - Used in simulation to verify behaviour, timing of Chip Layout.
- Process Design and Electrical Rule Check is performed before Chip is fabricated.
 - Highly detailed check done by foundry before fabrication

Future DSP Chip Designs

- Manchester Chip is only a trial run of the design flow.
- Future Chips
 - Will be fabricated in a smaller process, currently targeting a $0.18\mu\text{m}$ process
 - Will have greater functionality
 - Analogue and Digital
 - Will have more detailed simulation in the design flow
 - More timing analysis
 - Power analysis