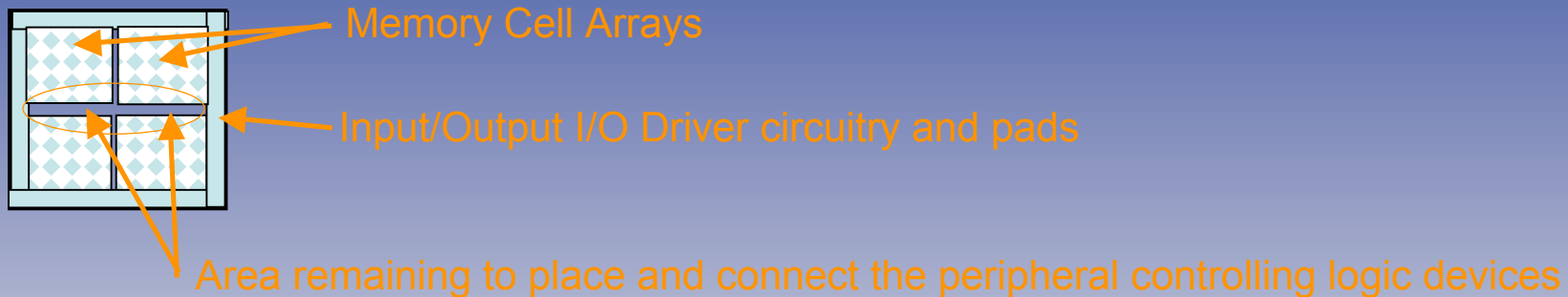# A design method for the automated layout of control logic for FLASH memory
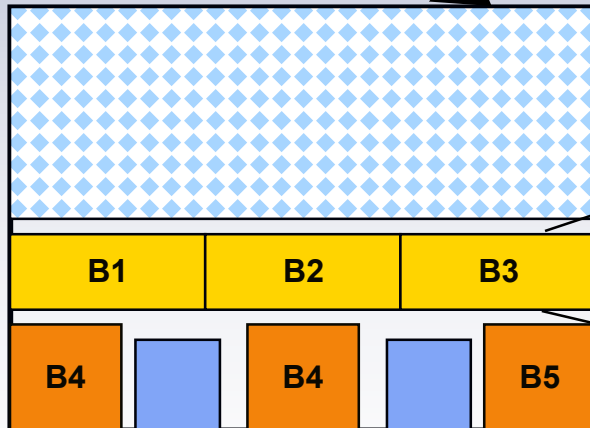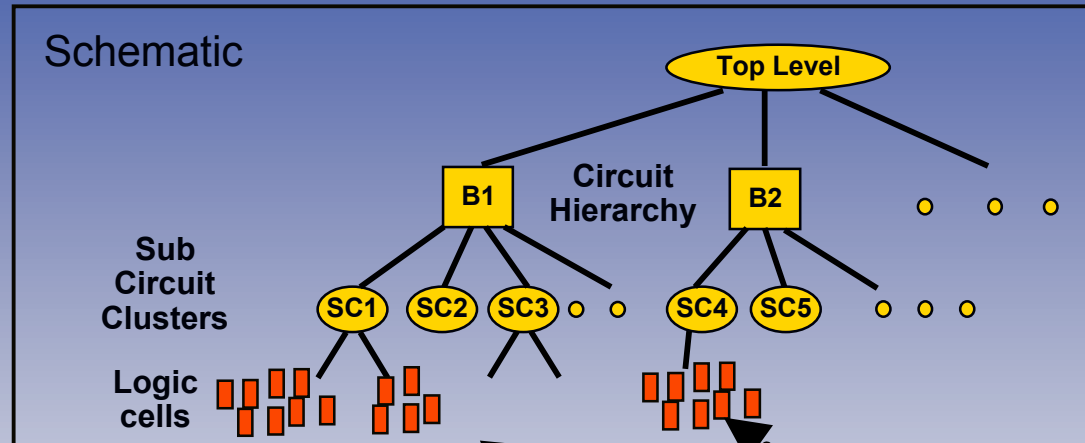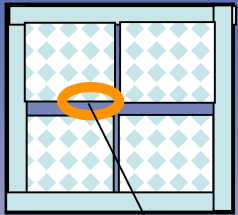
Kevin Steptoe Jeremy Birch

Pulsic Ltd

Key Issues for layout of peripheral logic for FLASH

Memory Cell Arrays

Input/Output I/O Driver circuitry and pads

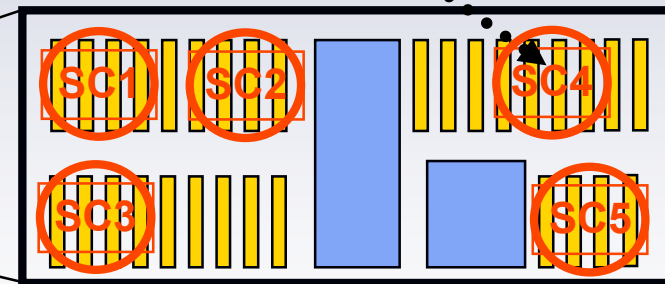Area remaining to place and connect the peripheral controlling logic devices

- Characteristics and requirements for peripheral logic layout for FLASH
  - Long and narrow aspect ratios of layout regions.
  - Need for control of specific layers (e.g high resistive layers)
  - Automation for low number of routing layers (e.g. 1-2 layer)
  - For area performance and yield very specific placement and routing styles are required.
  - nm process nodes are driving requirement for increased automation for productivity.
- Standard ASIC technology inappropriate and difficult to control to achieve required results.

Layout of Peripheral Logic – Objective is to physically place and connect logic cells
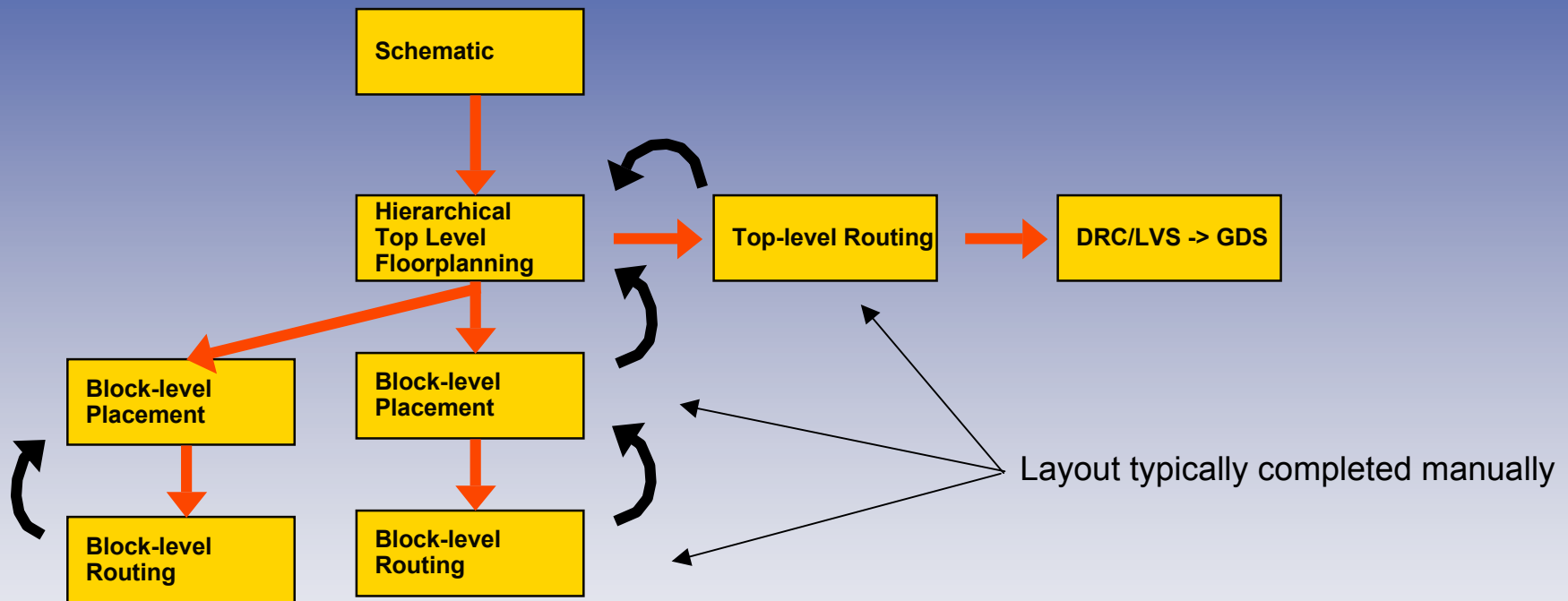

Schematic

Top Level

Circuit Hierarchy

B1   B2

Sub Circuit Clusters

SC1 SC2 SC3   SC4 SC5

Logic cells

Physical : Top level Floorplan

B1   B2   B3

B4   B4   B5

Block level layout
Place and Route (P&R)

SC1 SC2 SC4
SC3 SC5

Memory Design Existing Flows

```
                    ┌─────────────┐
                    │  Schematic  │
                    └─────────────┘
                           │
                           ▼
                    ┌──────────────┐        ┌──────────────────┐      ┌──────────────────┐
                    │ Hierarchical │   →    │ Top-level Routing│  →   │ DRC/LVS -> GDS   │
                    │ Top Level    │        └──────────────────┘      └──────────────────┘
                    │ Floorplanning│
                    └──────────────┘
          ┌──────────────┘     └──────────────┐
          ▼                                   ▼
  ┌──────────────┐                    ┌──────────────┐
  │ Block-level  │                    │ Block-level  │
  │ Placement    │                    │ Placement    │
  └──────────────┘                    └──────────────┘
          │                                   │
          ▼                                   ▼
  ┌──────────────┐                    ┌──────────────┐
  │ Block-level  │                    │ Block-level  │
  │ Routing      │                    │ Routing      │
  └──────────────┘                    └──────────────┘
```

Layout typically completed manually

→ Basic typical "disparate" design flow

→ Needed – 2 way flow with good feedback loop between tools

## Conclusion: Integrated and automated solution required

Block Level Placement (1)

- **Objective to place logic cells in Clusters**
- **Maintain hierarchical sub-circuit clusters**

Placement WITHOUT clustering of sub-circuits will result in longer net lengths & lower performance
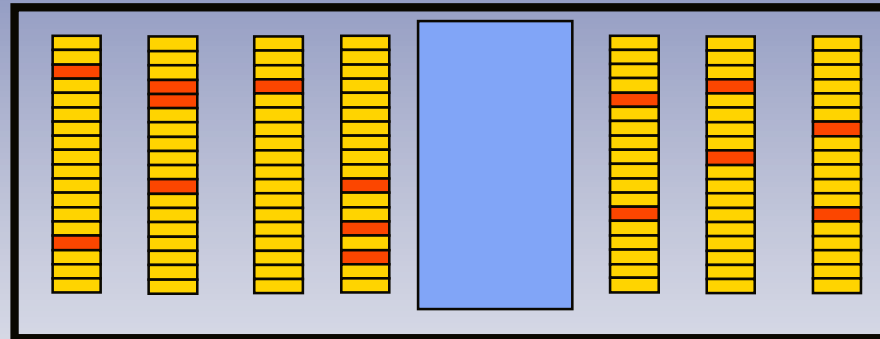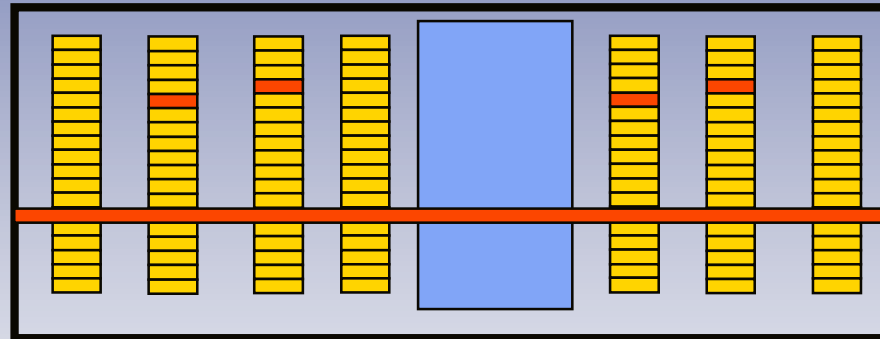
Placement WITH clustering of sub-circuit will result in shorter net lengths & increased performance
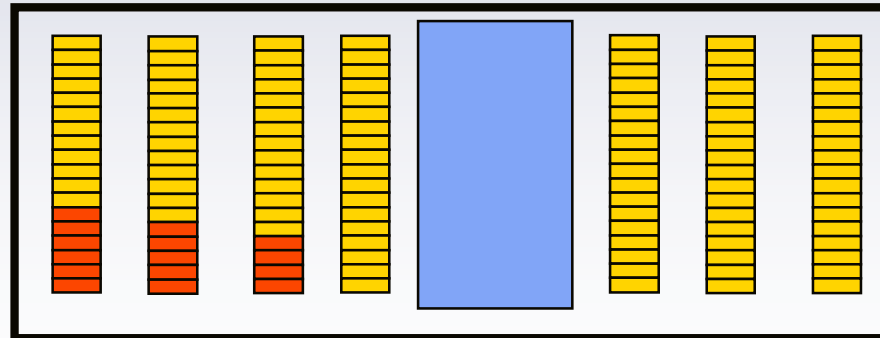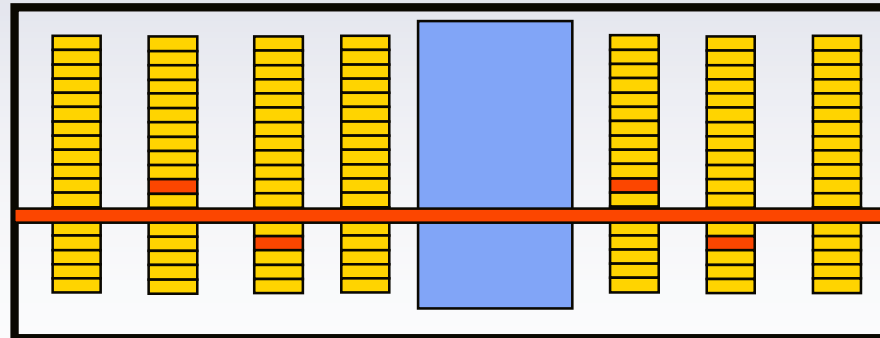
- Pre-Routing aware placement

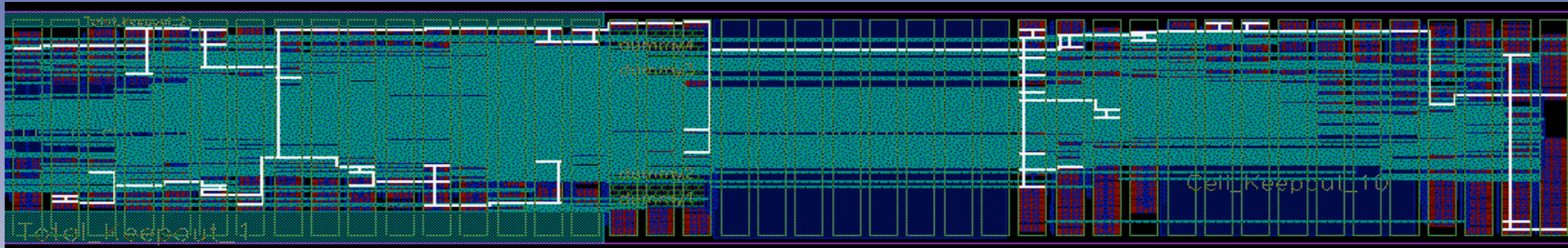**Without pre-routing awareness – critical net lengths can be unacceptably long**

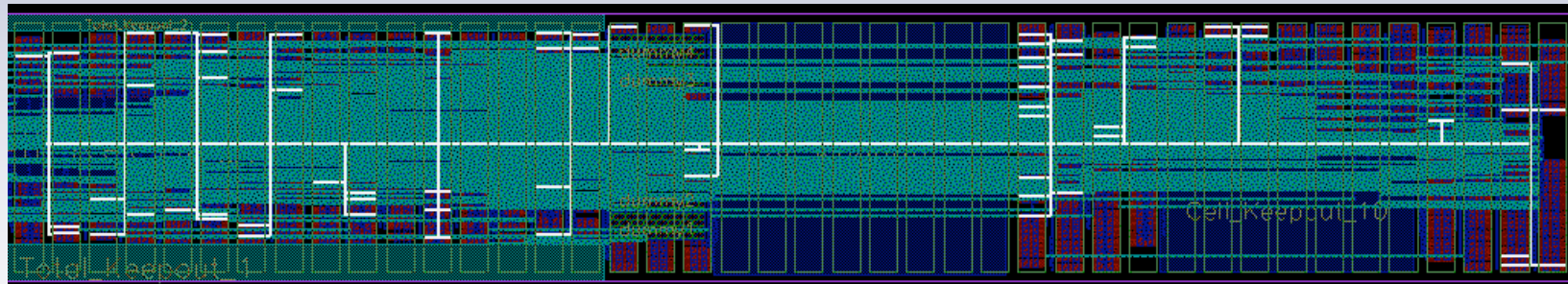**Placement that is Pre-routing aware will minimize total net length**

Block-Level Routing

❑ Normal autorouting approach will not achieve required results

Normal autoroute may min-connect the nets and be DRC correct – but with more bends & segments ➡

Spine & stitch route can achieve minimal bends and length to provide optimum performance & minimal area and still meet all DRC's ➡

Example of Routing Results

Without Methodology

With Methodology

Summary

Automation of the layout of peripheral logic is possible with a dedicated methodology involving specialised; floorplanning, placement and routing

- Integration of floorplanning together with placement and routing is mandatory
- Dedicated placement strategies are required for optimal results
    - Cluster placement
    - Routing aware placement
- Specialised routing techniques deliver smallest area, highest yield and best performance
    - Shape based routing
    - Spine and Stitch routing
    - DFM aware routing