

Datapath Floorplanning and Routing in a Three Metal Layer Process

Eric Marsman

Issues: The Horizontal/Vertical Convention for Metal Layers
Pitch Matching and Bitslice Width Matching

In EECS 427, one of the biggest challenges is to understand the entire micro-controller circuit design process before actually completing this process. This understanding is necessary because it provides a basis for making design decisions. In this document, an overview of datapath floorplanning and routing will be presented to facilitate visualization of the entire circuit layout and the associated pitfalls.

The Horizontal/Vertical Convention for Metal Layers

When routing metal lines on an integrated circuit, it is customary to associate a direction (horizontal or vertical) with each layer of metal. In a multi-layer metal process, the assigned directions typically alternate to allow easier routing. For example, in a five metal layer process, the convention would be to run metal 1, metal 3, and metal 5 horizontally while running metal 2 and metal 4 vertically. The EECS 427 micro-controller design uses a three metal layer process. For this discussion, metal 1 and metal 3 will be assigned the horizontal direction, and metal 2 will be assigned the vertical direction.

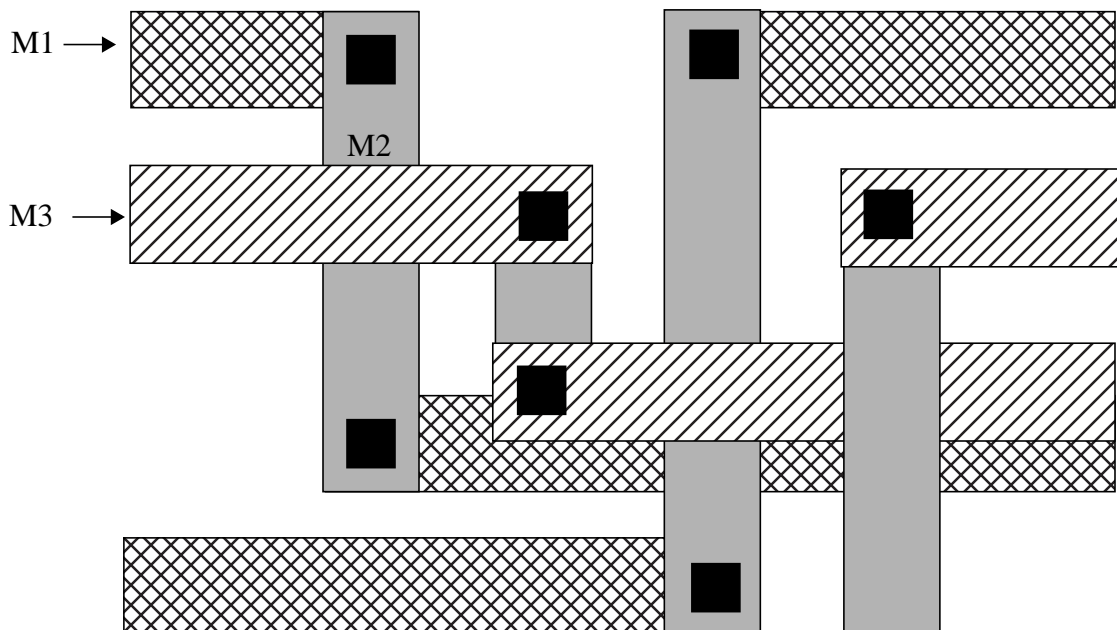


Figure 1: Ideal Routing Scheme

This convention is designed to make routing signals from one point on the IC to another point on the IC simple. Figure 1 shows how many signals “cross” over each another on their way to different destinations.

However, the routing process is made more complicated in EECS 427 by the limit of three metals. Not only do we need to route signals from point to point across the micro-controller, but we also must perform internal cell routing. This complication increases the complexity of the layout and requires more planning on the part of the designer, particularly regarding the routing of metal 1. Metal 1 is crucial because it has not only global horizontal routing responsibility, but also internal cell routing. Effective use of metal 3 for horizontal global routing can alleviate metal 1 routing conflicts.

To illustrate the basic challenges, a simple model for a 4 bit register is presented and coupled with a high-level representation of an ALU and a Shifter. In general, a register requires as its inputs and outputs the following signals: VDD, VSS, an input data bus, an output data bus, a clock, and a write enable. For a 4 bit register, the VDD, VSS, clock, and write enable signals are common to each of the 4 bits of the register. The input and output data buses, however, hold different values from bit to bit. This is the primary distinction between the two categories of signals. Figure 2 illustrates a possible routing scheme that fulfills the needs of the two types of signals and maintains a horizontal metal 1 and metal 3 and vertical metal 2 relationship on a global scale. Note that metal 1 is still used for internal cell routing, so the potential for it to run vertically exists.

In the Regfile, the write enable signal and the clock signal are shown running horizontally. In this setup, the signals will be delivered to each neighboring bit through metal 1 horizontally. Since the signals are common to each bit, this method of routing is appropriate. The bus signals, however, are routed in vertically in metal 2. These signals are common between multiple components with the same bit number, rather than across the component itself. Therefore, the vertical routing is preferred.

Pitch Matching and Bitslice Width Matching

Ultimately, the various components of the datapath will be assembled together and inter-connected. For this to happen smoothly, advance planning of the global routing scheme and component placement must be considered. Pitch matching and bitslice width matching are two techniques that will ease the placement and routing of the datapath components.

Pitch matching refers to maintaining a constant height between the power rails, VDD and VSS. As illustrated in Figure 2, pitch matching is performed individually for each component (Regfile, ALU, Shifter). This ensures that bits 0 through 3 of each component all line up horizontally. Note, however, that the pitch of the Regfile may be different than the pitch of the ALU or the Shifter.

On a global scale, bitslice width matching is implemented. Across all components, a constant bitslice width is maintained. This allows for easy routing of vertical (metal 2) bus interconnect throughout the datapath.

The horizontal routing will eventually be connected to vertical routing at the exterior edge of the datapath. As an example, VDD and VSS lines have been run vertically alongside bit 0. These allow the VDD and VSS lines from each of the components to be connected together and ultimately routed to pads.

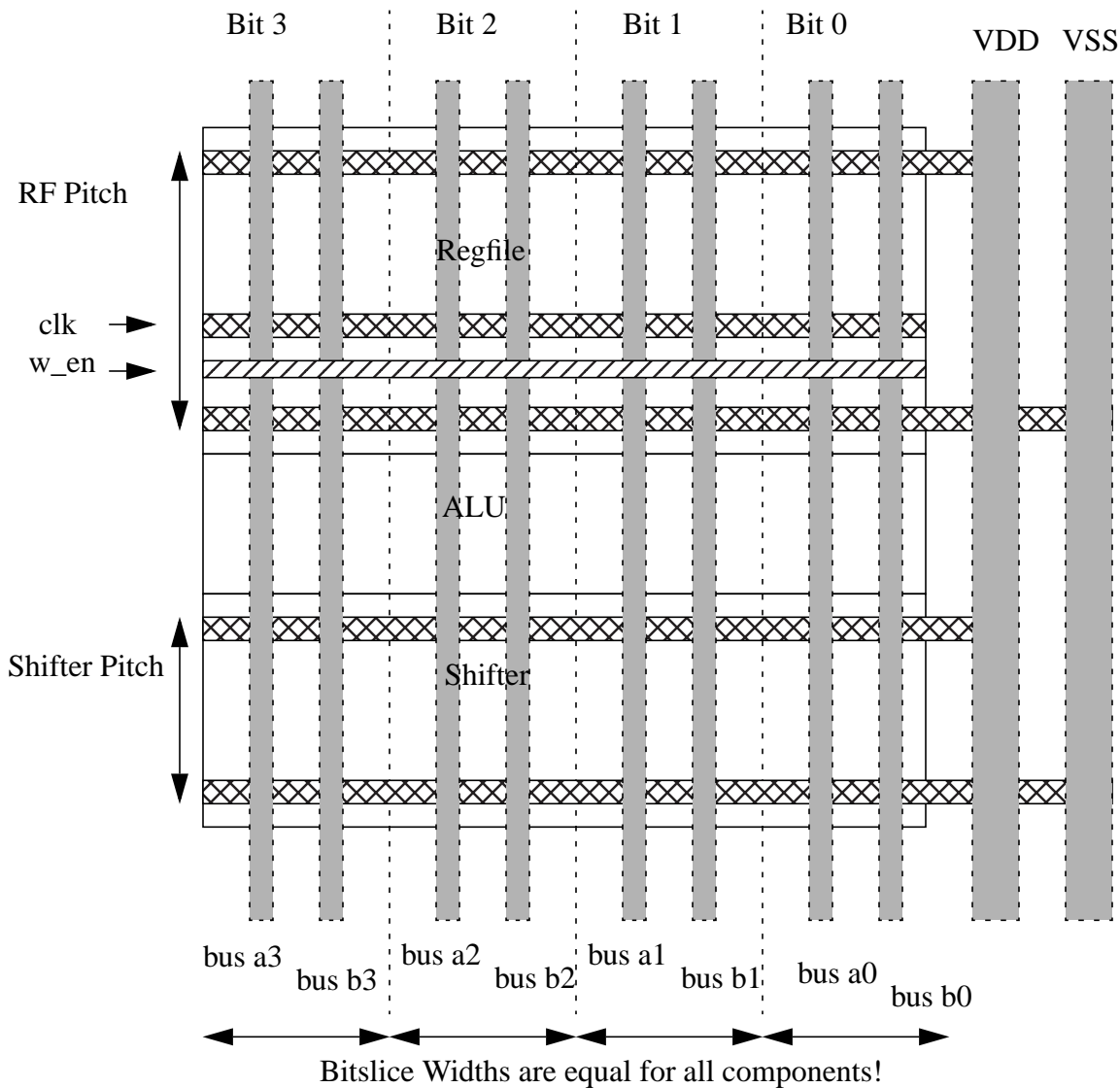


Figure 2: Example Datapath Routing Scheme (Partial)

Summary

The concepts presented should serve as a guide to help you visualize your circuit floorplan and provide you with some insight about the pitfalls of interconnection in a two metal layer process. This is merely one solution to the problem, and you are encouraged to think about possible alternatives that could yield potentially better results.