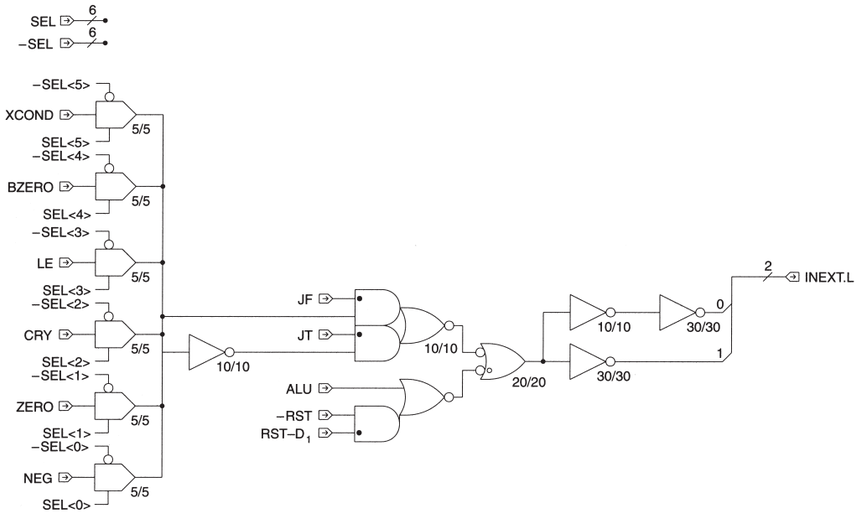

FIG 7.63 control—condition code selection

The notion of "well-formed" may differ from situation to situation, but a good starting point is the criteria placed on a 'well-formed" software subroutine. First of all, a well-defined interface is required. In the case of software this is an argument list with typed variables.

$$V_{gs3} = \sqrt{\frac{2I_{s3}}{\beta}} + V_t. \qquad \textbf{(2.56)}$$

$$I_{s4} = \frac{\beta}{2}\,(V_{gs} - V_t)^2 = I_{s3}.$$

Imagine that the power and chip size budget has been met by the four processor design that is outlined above. How should the design progress if it is discovered that significantly more processing power was required? There are two main directions in which to proceed. Firstly, all processors can be augmented in a uniform manner to improve each processor's performance.

b) pseudo-nMOS with pMOS transistors ¼ the strength of the pull-down stack

7 Sequential Circuit Design

7.1 Introduction

The manner in which one goes about designing a particular system, chip or circuit can have a profound impact on the effort expended on and outcome of the design. Over time, IC designers have developed and adapted design strategies from allied disciplines such as software engineering to form them into a cohesive set of principles that the novice designer can adopt to ensure timely, successful designs. This chapter will explore these principles.

"Multitudes of contrivances were designed, and almost endless drawings made, for the purpose of economizing the time and simplifying the mechanism of carriage."

—Charles Babbage, on Difference Engine No. 1, 1864 [Morrison61]

For the purposes of this chapter we will assume that an integrated circuit may be described in terms of three domains, namely: (1) the behavioral domain, (2) the structural domain, and (3) the physical domain. In each of these domains there are a number of design options that may be selected to solve a particular problem. For instance, at the behavioral level, the freedom to choose, say, a sequential or parallel algorithm is available. In the structural domain, the decision about which particular logic family, clocking strategy or circuit style is initially unbound. At the physical level, how the circuit is implemented in terms of chips, boards, and enclosures also provides many options for the designer. These domains may further be hierarchically divided into different levels of design abstraction. Classically, these have included the following for digital chips.

The relationship between description domains and levels of abstraction are elegantly shown by the Y-chart in Figure 7.1. In this diagram, the three radial lines represent the three description domains, namely the behavioral, structural and physical domains. Along each line are enumerated types of objects in that domain. Circles represent levels of similar design abstraction: the architectural, logic and circuit levels. The particular abstraction levels and design objects may differ slightly depending on the design method.

In this chapter we will examine the means by which we transform a description from one domain into a description in another domain while maintaining the integrity of the design. We begin by discussing some of the guiding principles that apply to most engineering projects. Then the various design strategies available the CMOS IC designer are surveyed, ranging from very fast prototyping or small volume approaches to approaches suitable for high volume digital, analog, or RF design. We examine the economics of design, which can guide us to the right selection of an implementation strategy and review