

# Overview of Digital Design

Dr DC Hendry

January 2006

## 1 Course Information:

1. Lecturer: Dr David Hendry.
2. Three meetings per week, normally two lectures and one tutorial.
3. Also laboratories, as advertised on the level 3 notice board.
4. Recommended texts: one of
  - Zwolinski “Digital System Design with VHDL” Pearson.
  - Dewey “Analysis and Design of Digital Systems with VHDL” PWS
  - Floyd “Digital Fundamentals with VHDL” Pearson.
5. Web Page: <http://www.eng.abdn.ac.uk/~eng186/eg3560>

## 2 Digital Systems

Digital electronics is at the heart of the majority of modern electronic devices - digital television sets, mobile phones, computers and game systems. Products such as automobiles are increasingly dependent upon digital electronic control systems to provide low cost implementations but also increasing functionality.

Digital systems, as opposed to analogue systems, have always been seen as providing excellent repeatability (each unit manufactured behaves as any other), tolerance to temperature variations and supply voltage variations, and a degree of noise immunity. In the early days of digital electronics however this came at considerable cost as discrete logic gates (individual NAND or NOR gates) were used to construct these systems. This resulted in physically massive systems - the early equivalent of todays PC filled a number of rooms. Representing data

with a 32-bit binary number simply required more components and wires than representing the same data with an analogue voltage.

The advent of *VLSI* (Very Large Scale Integration) devices however began to change this. Using an essentially photographic process it became possible to fabricate many thousands of logic gates on one piece of silicon - the *silicon chip*. Today we construct silicon chips with many millions of logic gates at low cost. For example, a 5mm x 5mm silicon chip will today contain about 3 million logic gates and costs about \$5 to manufacture in quantity. The design costs for such a device will normally exceed \$5M however. For smaller quantities, *FPGAs* may be used. These devices also provide logic gates, but logic gates whose function and interconnections may be chosen after manufacture by setting memory bits within the device. These are often economic for low to medium quantities, they are also very useful for experimenting and learning with.

The crucial point about today's digital design is the large number of gates involved in a design. Our design techniques must be able to permit completion of a design involving some millions of logic gates with an acceptable time-scale, and with a modest design team. Inevitably, design automation is adopted - that is, CAD tools (Computer Aided Design tools) are used.

The next section compares an earlier style of digital design based on schematic capture tools and manual design techniques with the current mainstream design technique based on simulation and *synthesis* of an *HDL* (Hardware Description Language).

## 3 Design Techniques

### 3.1 Manual Techniques

The very earliest digital circuits were of course designed with nothing more than pen and paper, and then debugged by building the circuit and checking with an oscilloscope and logic probe (a simple pen like device which indicates if the wire it touches is at logic '0', logic '1', or is not driven). As soon as computer systems with a graphics capability became available CAD tools were produced which permitted a design to be entered into the computer via *schematic capture*, and to be simulated.

When first introduced these tools greatly increased design productivity. Schematics were more rapidly entered and easily changed as the design progressed. Simulation tools permitted a degree of testing of the design before committing to a time consuming and costly construction phase.

As design complexity increased and commercial pressures required shorter and shorter design time-scales however, the limitations of this approach became clear.

Schematic entry tools required the designer to enter the design at the logic gate level. This does not mean that for a large design with one million logic gates a schematic with one million gates was required! For such a design *hierarchy* would be used to build up a larger design using many instances of a simpler design. Thus in principle a 32 bit adder would be constructed by designing a 1 bit adder as a schematic, and then constructing a 32 bit adder using 32 instances of that 1 bit adder. Nevertheless, the designer has first to manually design the logic circuit to gate level - a time consuming and tedious process.

At the lower levels of the design (expressed in terms of logic gates such as AND and OR) the schematic capture tool makes good use of graphics to provide an easily understood diagram. The logic symbols for elementary gates are standardised and easily recognised. At higher levels of the design (registers, functional units, FSMs) there are few standard symbols - the mnemonic value is lost.

## 3.2 The Hardware Description Language Approach

A Hardware Description Language (HDL) provides a computer like language which specifies the required digital logic. The designer uses this language to describe the system required. The design may be simulated and corrected as necessary. Next the designer *synthesises* the circuit to either an FPGA or to a VLSI device. This uses a computer tool analogous to a software compiler which translates the HDL description into a circuit consisting of basic logic gates and flip-flops as necessary. The resulting circuit is known as a *netlist*.

The synthesis process provides the designer with rapidly applied optimisation techniques, techniques which will usually be more effective than manual techniques. The ease of applying these techniques also means that the designer may have time to experiment with different design approaches at a higher level.

Use of the synthesis tool also means that when we prepare a design we do not have to commit to a particular manufacturer's FPGA or silicon process. It is only when we synthesise that a particular FPGA or silicon process must be specified. This has the great advantage that our HDL designs can be easily transferred from one process to another. Indeed, this capability has opened up a new industry segment known as *IP* (Intellectual Property). A number of companies now exist solely to provide *IP Cores*, HDL designs of for example a microprocessor, or perhaps an MPEG decoder. These companies then sell this synthesisable description to designers constructing complex designs.

## 4 Implementation Techniques

We've already mentioned FPGAs and VLSI devices above, this section gives a little more detail on these devices and one other means of implementing digital systems.

### 4.1 Discrete Components

The first digital systems were implemented with individual gates and flips-flops constructed from discrete transistors and other components. Indeed the first computer produced about 50 years ago used this techniques. Physically however these machines were large and very costly.

It was soon realised that rather than producing single transistor on a piece of silicon, packaging those individual transistors and then connecting them together on a printed circuit board, that it was very much more efficient to place a number of transistors forming a common circuit onto one piece of silicon - producing an *integrated circuit*. This gave rise to circuits forming complete logic gates within one package, for example, the 7400 device contains four two-input NAND gates in one package. These devices were described as *SSI* Small Scale Integration.

Until the availability of FPGAs for low volume products, and widespread access to VLSI design for high volume products, families such as the 74xx devices were the main implementation technique for digital circuits. As increasing numbers of transistors could be integrated onto a single chip, *MSI* (Medium Scale Integration) and *LSI* (Large Scale Integration) devices, devices such as the 741616 (a 16 bit x 16 bit multiplier) became available within this family.

### 4.2 VLSI Devices

VLSI (Very Large Scale Integration) devices included increasing numbers of transistors, and so logic gates, onto one piece of silicon. Initially, but a few companies produced these devices, companies such as Fairchild (regarding by many as the inventor of the integrated circuit), Texas Instruments, and National Semiconductor in the US, and Plessey, GEC and Ferranti in the UK.

These devices provided efficient implementations of commonly required functions. Designers then used a number of these VLSI devices, with SSI devices as "glue" to complete a design. This approach has limitations as to performance and cost.

The next step was to permit complete designs to be placed on a single chip. Since each design could be specific to a particular application, the terminology *ASIC* arose (Application Specific Integrated Circuit). The design of VLSI devices was now carried out in many more locations, although manufacture was concentrated in a few locations where the increasingly expensive fabrication plants were available. To cater for this design process great effort was expended on the research and development of CAD tools for VLSI devices.

Today's VLSI devices offer the designer many millions of logic gates at continually falling costs. There is one great disadvantage however. Once a design is in production unit cost may be say \$5, but the cost to design, including the cost of manufacturing the photographic masks required, will exceed \$5M. For small scale manufacture (and Universities!) this is a formidable barrier.

### 4.3 FPGAs

FPGAs (Field Programmable Gate Arrays) provide a VLSI device consisting of a large number of logic blocks whose function is determined by bits loaded into a memory within the device. These bits determine the function of each logic block, and the interconnect between. Loading these bit patterns is normally done when power is applied to the device by loading the *bit stream* from an adjacent memory chip.

Thus the manufacturer of the FPGA uses standard VLSI techniques to produce many millions of these devices. Lower scale manufacturers then load the required bit pattern (*program* the device) to give the circuit they require within their equipment.

## 5 Complexity

At the level of an individual gate, a digital circuit is very simple. Real circuits however do not consist of one or even a few gates, but of many gates (often millions) and flip-flops. The design problem then becomes one of dealing with the complexity which arises from a large number of basic components. Strategies for dealing with such complexity are:

### 5.1 Hierarchy

We've seen an example of this strategy already in section 3.1 where an example of building an adder using a large number of one-bit adders was used. This

same strategy can be used in a variety of forms, but the general pattern is the same.

To construct a larger component we consider a design in which that component consists of a number of smaller components each of which are identical. Repeated use of the simpler component then constructs the larger component.

## 5.2 Abstraction

When we build a digital system at different points in the design we think in terms of bits, or in terms of integers, or in terms of perhaps digital images. Clearly all integers in a digital system are eventually represented by bits, but we may not need that level of detail to think about a particular design problem. Abstraction levels refer to an appropriate choice of description to fit the particular design problem. Figure 1 illustrates a general set of abstraction levels for digital design. For a particular design however, abstractions appropriate to that design may also apply.

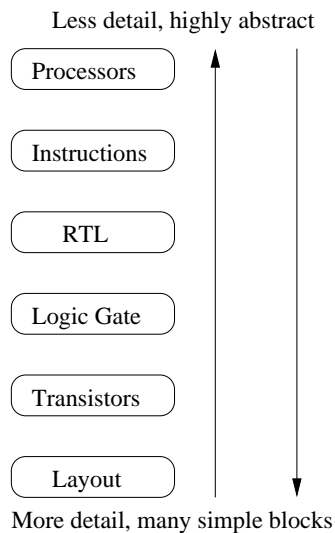


Figure 1: Abstraction Levels

This is not quite the same as hierarchy, as there may be layers of hierarchy within a given abstraction layer (for example, a flip-flop may be regarded as a number of interconnected logic gates).

## 6 Challenges

Both VLSI devices and FPGAs (really a subset of VLSI devices) are increasing in size year on year. Speed of execution increases, the number of devices available increases relentlessly it would seem. Table 1 gives an abstract from the ITRS (International Technology Roadmap for Semiconductors) predictions for this decade.

Year Introduced	1999	2002	2004	2008	2011
Feature Size <i>nm</i>	180	130	90	60	40
Transistors (Millions)	110	220	441	1852	4568

Table 1: Semiconductor Industry Roadmap - ASIC

Gate count in such a table is normally taken to be the number of transistors divided by four, and is then the equivalent number of 2-input NAND gates, since a 2-input NAND requires four transistors.

And of course, it's you who will have to design these devices! Once you have come to terms with the design process, the theoretical principles, the vast literature of methods available, and the CAD tools, you really do have enormous freedom - the main limitation is your imagination.