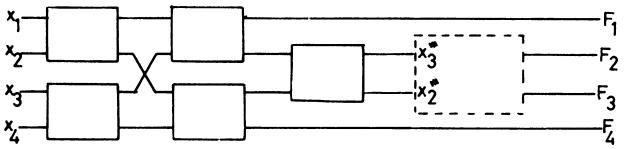

Fig. 16

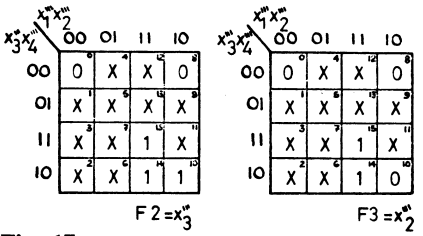

Fig. 17

$$x_1' = x_1 + x_2, \qquad x_2' = x_1 \cdot x_2, \qquad x_3' = x_3 + x_4$$
$$x_4' = x_3, x_4$$

This has the effect of *mapping* n-space $x_1 = 0$, $x_2 = 1$ *onto* $x_1 = 1$, $x_2 = 0$ and n-space $x_3 = 0$, $x_4 = 1$ *onto* $x_3 = 1$, $x_4 = 0$ as indicated in Fig. 11.

The n-spaces $x_1' = 0$, $x_2' = 1$ and $x_3' = 0$, $x_4' = 1$ may then be allocated the 'don't care' state 'X' since these n-spaces *cannot* be addressed by the inputs x_1, x_2, x_3, x_4. The four functions remaining to be synthesised after this step are shown in **Fig. 13.** These four functions also have common disjoint symmetries, viz. $x_1' \neq x_3'$ and $x_2' \neq x_4'$ (shown boxed and arrowed respectively in Fig. 13).

It should be noted that the importance of choosing *disjoint* symmetries is that propagation delays are minimised.

Fig. 14 shows the implementation of these symmetries by OR/AND modules viz.

$$x_1'' = x_1' + x_3', \qquad x_3'' = x_1' \cdot x_3',$$
$$x_2'' = x_2' + x_4', \qquad x_4'' = x_2' \cdot x_4'$$

The four functions remaining to be synthesised are shown in **Fig. 15.**

Now F1 is solved completely by letting F1 $= x_1''$ and F4 is solved completely by letting F4 $= x_4''$. Functions F2 and F3 have a common symmetry $x_3'' \neq x_2''$ (shown boxed in Fig. 15).

The implementation of this symmetry is shown in **Fig. 16**, viz

$$x_3''' = x_3'' + x_2'', \qquad x_2''' = x_3'' \cdot x_2''$$

The two functions (F2, F3) remaining are shown in **Fig. 17.** These are solved completely by letting F2 $= x_3'''$ and F3 $= x_2'''$. This gives the solution shown in Fig. 3(*b*).

In practice this procedure is carried out by a computer aided design method. Edwards (1977) should be consulted for a more detailed discourse on this design method.

References

BLATT, V., *et al.* (1974). Substrate Fed Logic—An improved form of injection logic, International Electron. Dev. Meeting (IEEE), *Technical Digest*, Washington DC, p. 511 fol.

EDWARDS, C. R. (1975(a)). The application of the Rademacher-Walsh transform to Boolean function classification and threshold logic synthesis, *IEEE Trans. Electron. Comput.*, Vol. C-24, pp. 48-62.

EDWARDS, C. R. (1975(b)). Some novel exclusive—OR/NOR circuits, *Electronics Lett.*, Vol. 11, No. 1, pp. 3-4.

EDWARDS, C. R. (1977). Design of Easily Tested Circuits using Mapping and Spectral Techniques, *The Radio and Electronic Engineer*, Vol. 47, No. 7, pp. 321-342.

HART, C. M., and SLOB, A. (1973). Integrated Injection Logic (I^2L), *Philips tech Rev.*, Vol. 33, No. 3, pp. 76-85.

HURST, S. L. (1973). Digital-Summation Threshold-Logic gates: a new circuit element, *Proc.IEE*, Vol. 120, No. 11, pp. 1301-1307.

REDDY, V. C. V., and SWAMY, P. S. N. (1974). Note on Digital-Summation Threshold-Logic gates, *Proc.IEE*, Vol. 121, No. 10, pp. 1085-1086.

Book review

Computer-Aided Design of Digital Systems, by D. Lewin, 1977; 313 pages. (*Edward Arnold*, £15·00)

This book will be of interest mainly to students of computer science at a postgraduate level, and to those practising engineers who are already familiar with 'formal' logic design methods. The book surveys the current status of computer aids in the fields of logic-network synthesis, logic simulation and logic testing. The subject of system specification, both by means of register transfer languages and by graph-theoretic models is also discussed.

The main barrier to the more widespread acceptance of computer aided logic design is the lack of sufficiently powerful algorithms, especially methods applicable to circuits using MSI and LSI components. Most of the algorithms described in this book are orientated toward design using flipflops and discrete NAND and NOR gates; this limits its usefulness to designers of CAD systems in industry, who have to work with the current technology.

The longest chapter in this book (117 pages) covers the topic of logic-network synthesis. A large number (over 20) of algorithms are described, for state reduction, state assignment and for implementation of the resulting switching functions in a particular 'logic family'. The algorithms described first appeared in a variety of journals: Ph.D theses, etc.; this book serves the useful purpose of collecting and comparing such a diversity of methods. The algorithms are described in Professor Lewin's usual lucid style, and a large number of helpful worked examples are provided. Many of these algorithms suffer severe limitations on the size of problem which they can handle. Unfortunately, little numeric information is provided in this book to indicate to the reader the limitations of each technique.

The chapter on system specification contains an up-to-date survey of hardware description languages, and also describes more recent developments, e.g. the use of Petri nets. The subjects of logic simulation for design verification and for test-program validation are treated in rather less detail; for example, the techniques of deductive fault simulation and the use of worst case timing are mentioned but not described in detail. The book concludes with a review of the subjects of logic-circuit testing and testable logic design. The book provides a large number (nearly 300) of references, and a useful subject and authors index.

D. BUMSTEAD (Poole)