Mask-programmable multiple-valued logic gate using resonant tunnelling diodes

H.L.E.Chan
M.Bhattacharya
P.Mazumder

Indexing terms: Logic gate, Resonant tunnelling diodes, Implementation

Abstract: The design of a 1-input and a 2-input 4-valued logic gate is described. These gates can be used to implement cells of a gate array, with the benefit of having programmable functionality and multiple-valued logic input and output lines. Programmable functionality allows any of the four billion functions of a 2-input 4-valued logic gate to be implemented by the same cell, and multiple-valued logic signal lines can reduce the area needed to be reserved for intercell routing. A key component of these cells is the resonant tunnelling diode, an ultrafast and small quantum electronic device that can be used in conjunction with HBTs to make compact literal circuits and decoders. So far as non-gate-array implementations are concerned, compactness can be improved further by a simple algorithm, also presented by the authors, to discard redundancies in the gate structure.

1 Introduction

In designing high-speed VLSI circuits, compactness has to be a major objective. A condensed circuit layout guarantees the minimisation of circuit delay as well as power dissipation as a direct consequence of reduced interconnect lengths. Long interconnects cause unwanted capacitances, excess power consumption and, in high-speed circuits, transmission line effects [1]. Weeding out excess interconnects can thus be regarded as the panacea for high-speed VLSI problems.

Multiple-valued logic (MVL) has been projected [2] as a powerful means of reducing the number of interconnects in a circuit by improving the signal encoding efficiency of each interconnect. Indeed, multiple-valued radix-r signal encoding enables a single line to carry $\log_2 r$ times the information conveyed by a binary connection. However, MVL circuits are difficult to design. The number of different combinations that are possible for an r-valued n-input gate is r^{r^n}. This means that there can be $4^{4^1} = 256$ different types of 1-input gates in a 4-valued system, and $4^{4^2} = 4\,294\,967\,296$ different 2-input gates in a 4-valued system. It is obviously impossible for a circuit designer to design so many different standard cells or gate arrays for a CAD library. An automatic generation or synthesis technique is needed.

Automatic synthesis of MVL circuits has received much attention in recent years. MVL circuits designed with CCD, CMOS, bipolar, or resonant tunnelling diodes (RTD) have been reported. However, these designs either require a large number of transistors to implement [3], or they require non-optimal heuristics for function decomposition [4, 5] or they violate the technological constraint for device integration (RTD and CMOS cannot be integrated together [4]). Furthermore, none of the MVL circuits reported can be used effectively for gate array implementations.

Gate arrays are a popular choice for low-to-medium volume IC manufacturing. The placements of different functional blocks and transistors of a gate array are fixed but routings are not done. These unrouted gate arrays can be fabricated in large quantities and the stockpile shared between different designs. To personalise these generic gate arrays, the designer needs only to make the mask for the final single metal layer or double metal layer with contacts and vias; this process is called mask-programming and can be done quickly. Lower prices (because of mass production) and faster turn around time (because only one or two masks are needed) make gate arrays an attractive IC manufacturing method.

This paper introduces the design of compact one- and two-input multiple-valued logic gates which can be mask programmed or be synthesised following simple rules. The mask programmable design is ideal for use in gate arrays not only because its MVL input and output lines require less space to route than its binary counterparts, but also because even the functionality of these gates can be changed at the last mask levels. The synthesisable design is an alternate version of the same gate for non-gate-array implementations. Both designs can be implemented with less transistors than other reported methods because of the use of RTDs.

2 Resonant tunnelling diodes and NDR-SPICE

Resonant tunnelling diodes are quantum electronic devices first proposed by Tsu and Esaki [6] in 1973. They can now be readily fabricated using molecular beam epitaxial (MBE) or chemical beam epitaxial (CBE) technologies in few dozen research institutions

IEE Proceedings online no. 19960571
Paper received 3rd July 1995

The authors are with the Department of Electrical Engineering & Computer Science, The University of Michigan, Ann Arbor, MI 48109, USA

[7]. RTDs have a folded (sawtooth) I-V characteristic (Fig. 1). At certain applied voltages, these diodes exhibit negative differential resistivity (NDR). This NDR characteristics are extremely valuable for designing MVL circuits. MVL multiplexers [8], memories [9], and adders [10] have been breadboarded with RTDs, and they show great promise for compactness and speed if made into integrated form. Circuits that exploit this NDR characteristic, though very promising, cannot easily be designed in the absence of a suitable simulator. First of all, conventional SPICE simulators do not have the models for these types of devices. Even if accurate device models are created, simulators will experience convergence problems whenever the sharp corners of the NDR characteristics are encountered. This is a well known problem in RTD simulation [11] which was solved in the past only by substituting the RTD model with an inaccurate one that does not have sharp corners in its characteristics. In 1994, Mazumder *et al.* [12] reported the development of a SPICE version called NDR-SPICE which can accurately represent and simulate any observable RTD (RHET, RTBT, MPRTD, HBT, HEMT, RTT, or BSRTT) characteristics. Models can be inserted through an I-V characteristics table and convergence problems are eliminated through memorising RTD states and through intelligent force convergence subroutines. All the simulation results in this paper are obtained through NDR-SPICE.

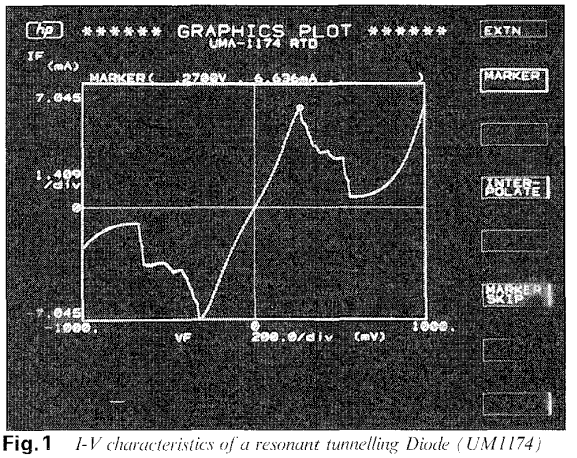

Fig. 1 *I-V characteristics of a resonant tunnelling Diode (UM1174)*

The models used for the RTDs and the HBTs (heterojunction bipolar transistors) are realistic level 1 SPICE models obtained from actual device measurements. Only npn HBTs are used in this design because pnp HBTs, though available, are not as fast switching as their npn counterparts. For maximum circuit performances, resistors, instead of pnp HBTs, are used as loads. Thin film resistors, made from nichrome, tantalum, or cermet [13] with sheet resistances in the range of 10–2500Ω/□ can be used to deposit the right resistances in an area comparable to that of a transistor. In some cases, the resistors can also overlap the transistors to some extent, making this RTD-HBT-resistor based MVL design potentially very compact.

3 One-input multiple valued logic gate

An r-valued, n-variable function $f(X)$, where $X = \{x_0, x_1, ..., x_{n-1}\}$ with x taking on values from $R = \{0, 1, 2, ..., r-1\}$, is a function $f: R^n \Rightarrow R$ which maps n-inputs of radix-r numbers into one output of radix-r number. The 1-input 4-valued logic gate described in this Section is a circuit that realises such a function in the voltage domain.

Here we have to define another function $g(Y)$ to describe each individual gate. A gate $g(Y)$, where $Y = \{y_0, y_1, ..., y_{r-1}\}$, is a gate that maps $i \Rightarrow y_i$. For example a 4-valued cycle gate can be described by $g(1, 2, 3, 0)$ (see Fig. 2).

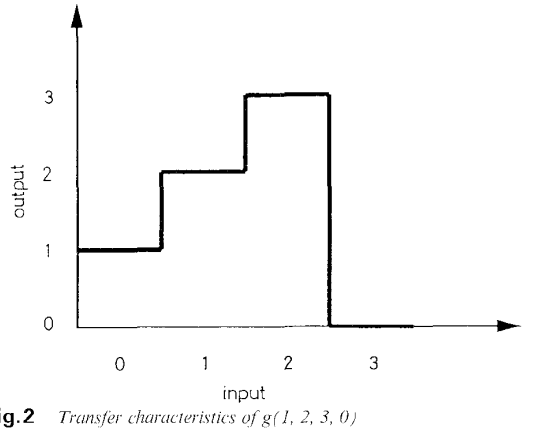

Fig. 2 *Transfer characteristics of g(1, 2, 3, 0)*

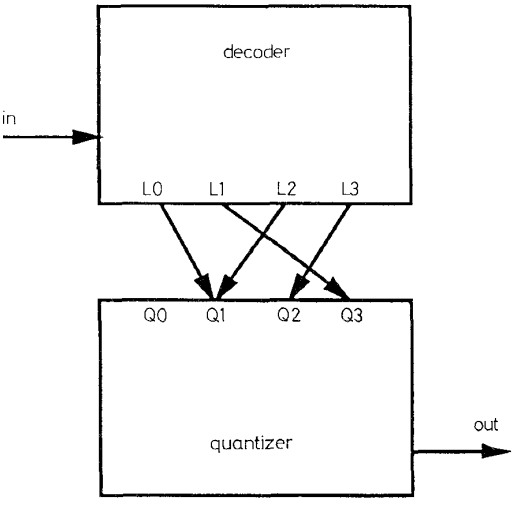

Fig. 3 *Architecture of a 1-input mask-programmable 4-valued logic gate*

The structure of the proposed programmable 1-input 4-valued logic gate is shown in Fig. 3. The decoder is a circuit that will generate a high in one of its four outputs (L-lines) based on the input's logic level. The quantiser is a circuit that will convert a high on one of its inputs (Q-nodes) to a predefined voltage level. Together, the gate can be mask-programmed by connecting each L-line to some Q-node. For example, to personalise a $g(1, 0, 2, 3)$ gate, one can connect $L0 \rightarrow Q1$, $L1 \rightarrow Q0$, $L2 \rightarrow Q2$, $L3 \rightarrow Q3$ through a metal layer, and to personalise a $g(1, 3, 1, 2)$ gate, connecting $L0 \rightarrow Q1$, $L1 \rightarrow Q3$, $L2 \rightarrow Q1$, $L3 \rightarrow Q2$ will suffice (Fig. 3).

3.1 Decoder

The decoder is made up of four literal circuits each turning on at a different non-overlapping voltage

range. Literals can be implemented very efficiently using RTDs. To understand how that can be done, we have to first understand the operation of an RTBT (resonant tunnelling bipolar transistor) inverter. It is a vertical integrated structure of RTD and HBT, with the emitter of the HBT grown directly on top of an RTD. An RTBT connected to an emitter and a collector resistor according to Fig. 4 forms an RTBT inverter. The transfer characteristics of an RTBT inverter are shown in Fig. 5.

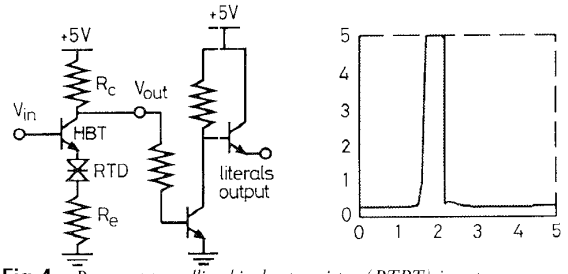

Fig.4 *Resonant tunnelling bipolar transistor (RTBT) inverter*
The graph shows literal output in volts against V_{in} in volts

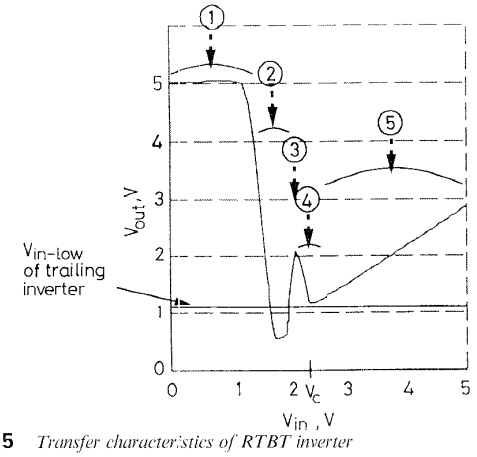

Fig.5 *Transfer characteristics of RTBT inverter*

The transfer characteristics of an RTBT inverter can be better explained if we consider the several regions of operation of the inverter separately. In region 1, $V_{in} < V_{sat}$ and the transistor is cut off, no current flows through R_c and so $V_{out} = V_{cc}$. In region 2, $V_{in} > V_{sat}$ and the HBT is in saturation, V_{out} thus goes low like a normal inverter. In region 3, the voltage across the RTD has exceeded V_p (the peak voltage of the RTD), the current across the RTD thus drops drastically because of the negative differential characteristics of the RTD; this limits the current across R_c also and so V_{out} jumps up. In region 4, the RTD is in its second positive differential resistance region and increasing the voltage across the RTD allows more current to flow, V_{out} thus decreases again. In region 5, the transistor is basically in saturation, and V_{out} is pushed up by the voltage at the RTBT's emitter which increases because of the emitter resistance.

To make this RTBT inverter into a literal circuit, we need to connect V_{out} to a normal inverter. R_c has to satisfy the following inequality:

$$R_c > \frac{V_{cc}}{I_p(of RTD)}$$

to allow V_{out} to drop below the turn-on voltage of the next stage inverter before going into region 3 described above. It is also important that R_c be large enough to prevent a second literal from being generated at V_c. The output of the literal circuit is shown in Fig. 6.

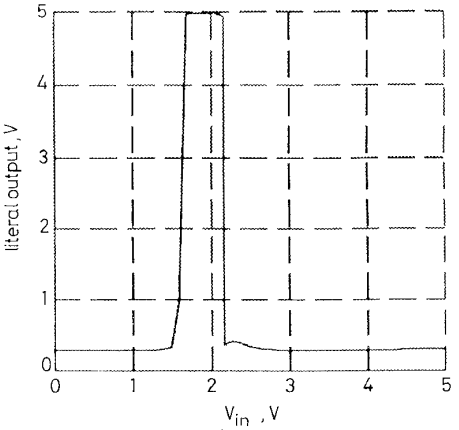

Fig.6 *Output of the RTBT literal circuit*

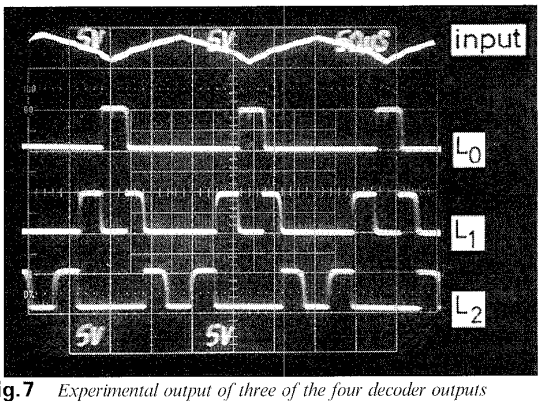

Fig.7 *Experimental output of three of the four decoder outputs*

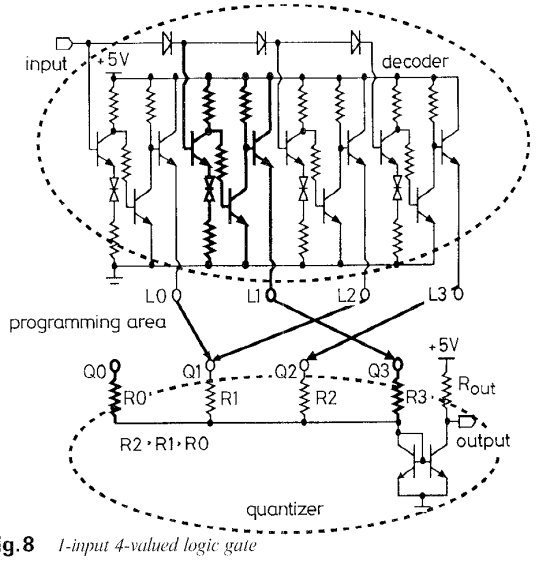

Fig.8 *1-input 4-valued logic gate*

Transistors connected to the output of the literal circuit in an emitter follower configuration not only improve the current drive of the output, but also act as a MAX circuit to allow multiple L-lines to connect to a single Q-node (e.g. $L0$ and $L2$ in Fig. 8).

A decoder is formed when the literal circuits are

connected together at their inputs through a chain of diodes. The diode chain separates the node voltages by intervals of V_d. Identical literals connected to different nodes thus turn on at different applied voltage intervals. Fig. 7 shows the experimental output of the decoder.

3.2 Quantiser

The quantiser is a current mirror with multiple input resistors of different values. The fact that the output of only one decoder is high and all the others are reverse biased at any one time simplifies the design of the quantiser. Such a quantiser can usually be designed through the simulator by fixing R_{out} and varying the input resistance ($R0$–$R2$) until a high in the corresponding Q-node generates the desired voltage level at the output. The process is repeated for all the Q-nodes with different desired output voltage levels in mind.

In most designs, logic level 3 corresponds to V_{cc}. This is the default output voltage of the quantiser if no Q-nodes are high. The resistance corresponding to $Q3$ can therefore be omitted (infinite resistance or open). In non-gate-array implementations, as will be explained in greater detail in Section 5, any literal circuit of the decoder connected to $Q3$ can also be eliminated because of this open circuit, allowing the one-input MVL gate to occupy even less area.

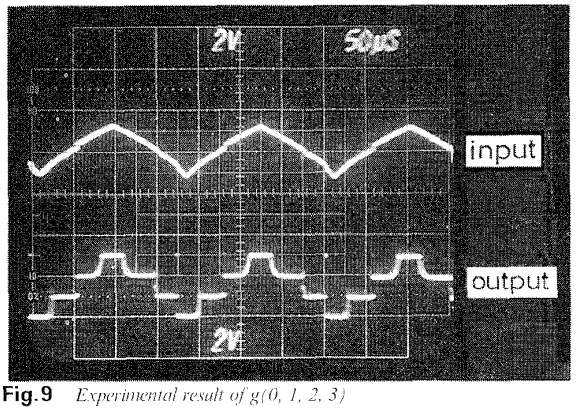

Fig. 9 *Experimental result of g(0, 1, 2, 3)*

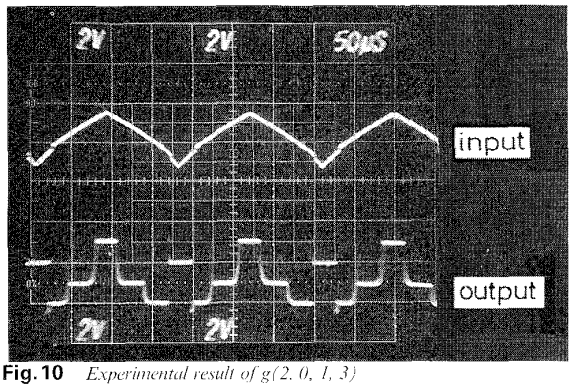

Fig. 10 *Experimental result of g(2, 0, 1, 3)*

A one-input MVL programmable gate has been breadboarded with 2N2222 transistors and RTDs fabricated in the University of Michigan (UM1174). Fig. 7 shows the input and three outputs of the decoder. It can be seen that these outputs are non-overlapping and shifted with respect to each other, just as designed. Outputs of gates programmed as $g(0, 1, 2, 3)$, $g(2, 0, 1, 3)$ and $g(1, 2, 3, 0)$ together with their inputs are also shown in Figs. 9–11. They serve as examples for the 256 different possibilities programmable with this gate.

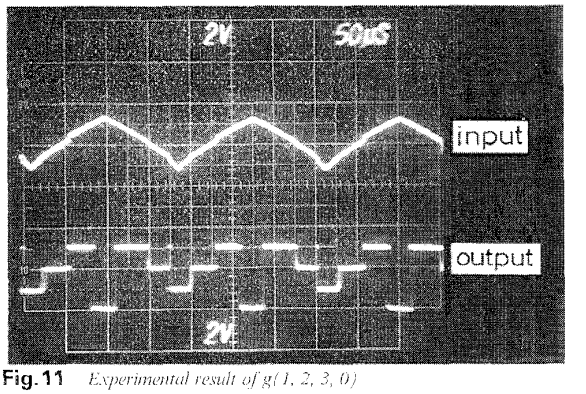

Fig. 11 *Experimental result of g(1, 2, 3, 0)*

IN1 \ IN0	0	1	2	3
0	3	2	3	2
1	1	3	1	3
2	3	0	0	1
3	3	0	3	2

Fig. 12 *4-valued logic Karnaugh map*

4 Two-input multiple-valued logic gate

A similar design can be used to implement 2-input MVL gates. The functionality of a 2-input gate is best described by its MVL Karnaugh map. An r-valued 2-input Karnaugh map has r rows and r columns, and each cell entry can assume any one of the r possible logical values. Fig. 12 shows an example Karnaugh map for a 2-input 4-valued logic gate.

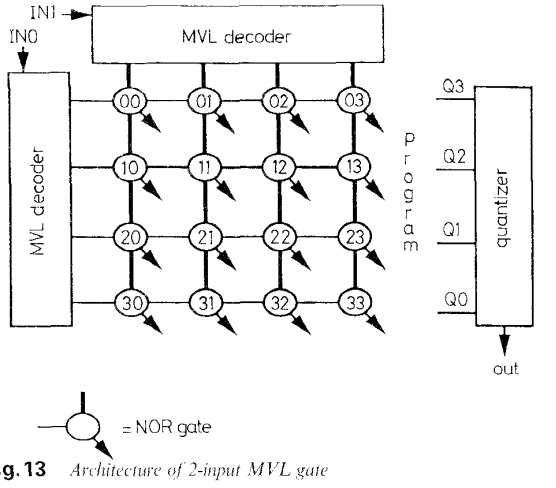

Fig. 13 *Architecture of 2-input MVL gate*

The structure of the gate is given in Fig. 13. Here negative decoders and NOR gates are used because

they are easy to implement. A negative decoder is a circuit that has four outputs which will go 'low' at non-overlapping but adjacent voltage ranges. 'Low' is defined here by the input of the NOR gate that is connected to it. The complete circuit is shown in Fig. 14.

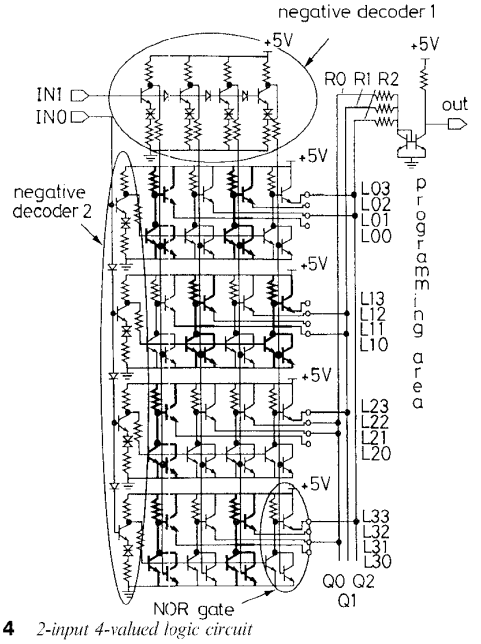

Fig. 14 *2-input 4-valued logic circuit*

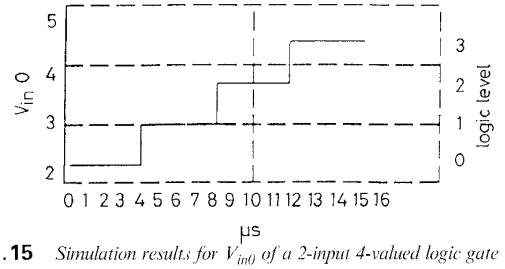

Fig. 15 *Simulation results for V_{in0} of a 2-input 4-valued logic gate*

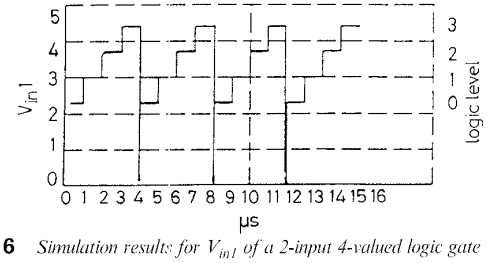

Fig. 16 *Simulation results for V_{in1} of a 2-input 4-valued logic gate*

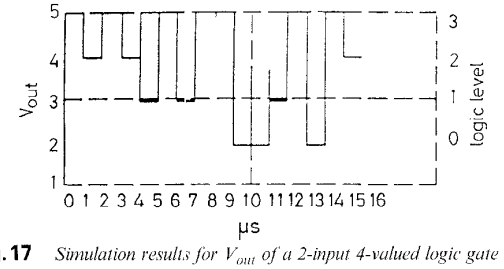

Fig. 17 *Simulation results for V_{out} of a 2-input 4-valued logic gate*

NDR-SPICE simulation for the circuit realising the function specified by the Karnaugh map described in Fig. 12 is shown in Figs. 15–17. The performance of this circuit, so far as speed and power dissipation are concerned, is presented in Table 1. All simulations were performed at a supply voltage of 5V.

Table 1: Performance of the 2-input 4-valued logic gate

Circuit	Number of transistors	Speed, MHz	Power, mW
Original circuit	58	250	680
Minimised circuit	37	340	529
Reduced capacitance HBT model	37	2000	527

5 Synthesis

Gate array implementation of logic functions is done by interconnecting a fixed array of logic gates as required. While this approach towards VHSIC (very high speed integrated circuit) manufacturing is very effective in dealing with the challenge of design complexity, it may incur unacceptable compromises so far as circuit performance is concerned [14].

To obtain the best possible performance from an IC, custom design should be performed. The advantages of custom design over gate arrays are manifold. In custom logic, we do not have to reserve channels for the interconnections, hence circuit blocks can be packed closer together. This leads to reduced area as well as reduced stray capacitances. A more important advantage is the customisation of individual logic gates. This may include choosing optimal transistor sizes, standing currents, number of emitters and collectors and omission of redundant circuit parts [14].

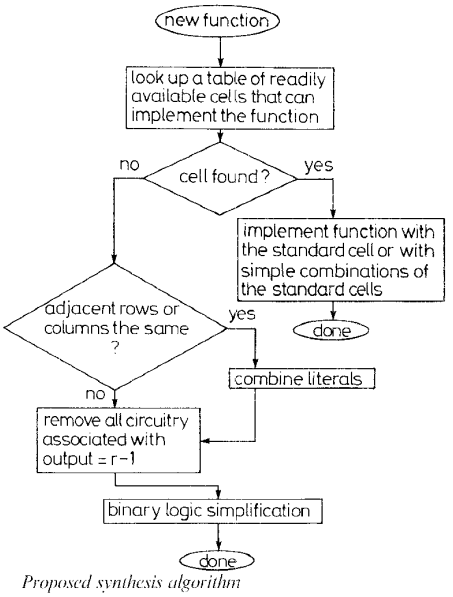

Fig. 18 *Proposed synthesis algorithm*

In custom chip design, combinational logic blocks can be generated by means of a well developed synthesis algorithm. In the past, synthesis of MVL has not been very successful. Unlike switching functions, an MVL function can often be realised by different combinations of product terms and summing operators. For example, Vranesic *et al.* [15] used min, max, and literal operators, and McCluskey [16] used min, tsum, and literal operators. Limiting the type of

operator is not beneficial in searching for the optimal (in terms of transistor count) implementation of a function. On the contrary, it limits the choices and flexibility that can be applied. This constraint is unfortunately required by most MVL synthesis algorithms. A much better way to synthesise would be to use the scheme of Fig. 18 which uses an existing well-developed binary simplification algorithm.

This MVL gate design methodology allows transistor count reduction under many situations. Some of these cases are discussed below.

(1) Some trivial functions, for example, $g(0, 0, 0, 0)$ in the one-input MVL gate case, can obviously be replaced by ground or some voltage source that supplies a constant voltage that corresponds to the logic level 0. Some other functions, of the type $g(0, 3, 0, 0)$ can be realised by a single literal circuit, which can be implemented very simply with the circuit shown in Fig. 4. Functions of the form $g(0, 3, 0, 3)$ can be implemented by two literals whose inputs are separated by an appropriate number of diodes. These and many other special case circuits can be predesigned and their functions stored in a table for the synthesis tool to look up. Other functions, that cannot be realised so easily, can then pass through the synthesis procedure.

(2) If two adjacent rows or columns of a 2-input 4-valued logic Karnaugh map are identical, then the hardware required for one of the rows or columns can be completely eliminated. This is because literals can be made (by reducing R_c in Fig. 4) to correspond to more than one input. A single 'wide' literal can thus replace two 'narrower' adjacent literals in the decoder and save all associated circuitry. ('wide' and 'narrow' refer to the width of the literal characteristics, see Fig. 6).

(3) After combining literals, circuits associated with output = $r - 1$ should be removed. In an r-valued logic system, logic value $r - 1$ can be represented by V_{cc}, which is the default output of the quantiser as mentioned in Section 3.2. This is done to the circuits in [8, 14], where optional hardware is shown in bolder lines.

The last step of the synthesis process is only applicable to two or more input MVL gates. Each different output (0, 1, ..., $r - 2$ in an r-valued logic gate) is independently synthesised using NAND-NOR gates as functional building blocks. The inputs are the binary literal outputs of the decoder and the output of this layer are the Q-nodes of the quantiser. This type of synthesis is well established and can be done relatively fast in case thousands of synthesis have to be done together for a VLSI system.

6 Conclusions

The design of a 1-input and a 2-input 4-valued logic gate has been described in this paper. The design principle of these gates can be applied in general to other n-input r-valued logic gates as well. The use of resonant tunnelling diodes in the circuit allows literals to be implemented very efficiently. These literals are in turn used to build decoders and then mask-programmable MVL gates.

The MVL gate described in this paper can either be used in gate-arrays to provide a compact and flexible way of implementing combinational logic, or as a basic structure for logic synthesis. Adopting this design in gate-arrays is especially promising not only because the gate's multiple-valued input and output signals can reduce the number of interconnections needed for intergate routing, but also because the functionality of each gate can be changed even upon the final metal mask layers.

So far as synthesis is concerned, this design has the benefit of being able to utilise binary simplification methods that are readily available and optimised for different technologies. What this paper suggests is an algorithm to presimplify the MVL gate design even before binary simplification method is applied.

7 Acknowledgments

This work was supported by the US Army Research Office under the URI program contract and by ARPA under contract DAAH04-93. The authors also thank Prof. Haddad, Dr W.L. Chen and Mr A. Gonzalez for their comments.

8 References

1 KAMEYAMA, M.: 'Toward the age of beyond-binary electronics and systems', IEEE 20th international symposium on *MVL*, IEEE, 1990, pp. 162–166
2 SMITH, K.C., and GULAK, P.G.: 'Prospects for multiple-valued integrated circuits', *IEICE Trans. Electron.*, 1993, **E76–C**, pp. 372–382
3 HANYU, T., YABE, Y., and KAMEYAMA, M.: 'Multiple-valued programmable logic array based on a resonant-tunnelling diode model', *IEICE Trans. Electron.*, 1993, **E76–C**, (7), pp. 1126–1132
4 DENG, X., HANYU, T., and KAMEYAMA, M.: 'Design and evaluation of a current-mode multiple-valued pla based on a resonant tunnelling transistor model', *IEE Proc. Circuits Devices Syst.*, 1994, **141**, pp. 445–450
5 ABD-EL-BARR, M.H.: 'Programmable synthesis of multi-valued multi-threshold functions for implementation using charge-coupled devices', *Int. J. Electron.*, 1992, pp. 345–370
6 TSU, R., and ESAKI, L.: 'Tunnelling in a finite superlattice', *Appl. Phys. Lett.*, 1973, **22**, p. 562
7 CHEN, W.L., MUNNS, G., EAST, J.R., and HADDAD, G.I.: 'InGaAs/AlAs/InGaAsP resonant tunnelling hot electron transistors grown by chemical beam epitaxy', *IEEE Trans. Electron Devices*, 1994, **41**, (2), pp. 155–161
8 CHAN, E., MOHAN, S., MAZUMDER, P., and HADDAD, G.I.: 'Multivalued multiplexer design using resonant tunnelling devices and heterojunction bipolar transistors', *IEEE J. Solid-State Circuits*, 1996, **31**, pp. 1151–1156
9 WEI, S.J., and LIN, H.C.: 'Multiple peak resonant tunnelling diode for multi-valued memory', *IEEE Trans. Electron Devices*, 1992, **39**, pp. 190–195
10 MICHEEL, L.J., and PAULUS, M.J.: 'Differential multiple-valued logic using resonant tunnelling diodes'. IEEE 20th international symposium on *MVL*, IEEE, 1990, pp. 189–195
11 MOHAN, S., SUN, J.P., MAZUMDER, P., and HADDAD, G.I.: 'Device and circuit simulation of quantum electronic devices', *IEEE Trans. Computer-Aided Des. Integr. Circuits Syst.*, 1995, **14**, pp. 653–662
12 MAZUMDER, P., SUN, J.P., MOHAN, S., and HADDAD, G.I.: 'DC and transient simulation of resonant tunnelling devices in NDR-SPICE'. 1st international symposium on *Compound semiconductors*, 1994
13 WOLF, S.: 'Silicon processing for the VLSI era, vol. 2: process integration, vol.2' (Lattice Press, California, 1990), pp. 731–736
14 BARNA, A.: 'VHSIC technologies and tradeoffs' (Wiley & Sons, 1981)
15 VRANESIC, Z.G., LEE, E.S., and SMITH, K.C.: 'A many-valued algebra for switching systems', *IEEE Trans.*, 1970, **C–19**, pp. 964–971
16 McCLUSKEY, E.J.: 'Logic design of multivalued logic circuits', *IEEE Trans.*, 1979, **C–28**, pp. 546–559