Approximate Pattern Matching in Nanotechnology

Viswanath Annampedu* and Meghanad D. Wagh**

* Storage Products, Agere Systems Inc., Allentown, PA 18109, vannampedu@agere.com
** Dept. of ECE, Lehigh University, Bethelehem, PA 18015, mdw0@lehigh.edu

ABSTRACT

Approximate pattern matching is comparing an unknown pattern with a database of stored patterns with a specified error tolerance. This ability to compensate for real world sensor errors makes approximate pattern matching an ideal choice for a wide range of applications including fingerprint, picture and bar-code identification, industrial automation, robotics and bioinformatics. This paper shows that a target pattern may be matched with a library of patterns within any specified error tolerance with a single deterministic threshold function. This function for n bit patterns can be implemented in nanotechnology using $2n+1$ unit area RTDs and one RTD of 1.5 times unit area and is configurable for any error tolerance and any given target pattern.

keywords: approximate pattern matching, threshold circuits, RTD, nanotechnology, reconfiguration

1 INTRODUCTION

Pattern matching systems usually look for a match between an unknown pattern and a database of stored patterns. Approximate pattern matching systems go one step beyond. They allow for patterns which differ from target pattern in at most e positions, where e is a preset error tolerance. Approximate pattern matching is better suited to real world applications where some error tolerance is highly desirable to compensate for inevitable sensor errors. These problems arise in a wide range of applications including identification of fingerprints, pictures or bar-codes against target patterns, detection of certain genes within DNA strands and consumer electronics such as home security devices, intelligent toys and robots. Most such applications demand high speed, reconfigurability to deal with a multitude of target patterns, error tolerance and low complexity and cost.

The requirements specified above suggest use of dedicated hardware architectures for approximate pattern matching. Such hardware architectures can be either sequential or combinatorial. Sequential systems are generally less complex, but fail to take advantage of many detection technologies such as Charge Coupled Device (CCD) arrays which inherently present data in parallel form. In addition, most high speed dedicated circuits for approximate pattern matching are married to a single pattern. Thus they cannot be easily reconfigured for another target pattern. Some of the hardware solutions to approximate pattern matching are presented in [1]–[5].

This paper investigates ultra-fast approximate pattern matching systems which can tolerate a preset number of errors. We show that the problem can be solved by single stage threshold circuits which may be easily implemented by nanotechnology architectures based on resonant tunneling diodes. The hardware complexity of the resultant architectures to match n bit patterns (within any error tolerance) is $O(n)$. This nanotechnology implementation implies high speed, low cost, low power and small size. A significant amount of research is already available in the technology of nanodevices. This paper puts focus on a novel application of these devices and bridges the gap between device physics and practice.

This paper is organized as follows. In section 2 we define and discuss threshold functions. We then show that the approximate pattern matching system can be viewed as a threshold function. Section 3 introduces the resonant tunneling diodes (RTDs) and the concept of a MOBILE architecture. We also implement the approximate pattern matching circuit using RTDs. Sec. 4 extends the architecture to allow for reconfigurability. This new architecture can be configured to match any target pattern against a library of patterns with any desired error tolerance. The architecture to match n bit patterns uses $2n+1$ RTDs of unit area and one RTD of 1.5 times the unit area. Finally, Section 5 presents the concluding remarks.

2 PATTERN MATCHING AND THRESHOLD LOGIC

A function $f(x_1, x_2, \ldots, x_n)$ is called a threshold function [6] if one can determine weights $w_1, w_2, \ldots, w_n$ and threshold T (all real numbers) such that

$$\sum_{i=1}^{n} w_i x_i \geq T \text{ if and only if } f(x_1, x_2, \ldots, x_n) = 1. \quad (1)$$

The representation of the threshold function is shown in Fig. 1. As an illustration, one can verify that $f(a, b, c) =$

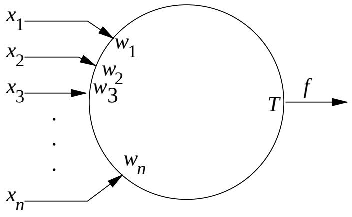

Figure 1: Representation of the threshold function f.

$a+\overline{b}c$ is threshold with weights 2, -1 and 1 and a threshold value of 0.5. Note that these weights and thresholds are not unique.

Some basic gates such as AND, OR and NOT are threshold functions. But Exclusive-OR gate or simple functions such as $ab + cd$ or $ab + \overline{a}c$ cannot be realized as single threshold functions. In general, larger Boolean functions are rarely threshold. Further, it is very hard to determine if a function is threshold or not. The machinery of k-monotonicity often used to establish threshold nature, is quite elaborate for most engineering settings [6]. However, we now show in the following theorem (proof omitted for brevity) that the Boolean function corresponding to the approximate pattern matching is indeed a threshold function.

Theorem 1. The logic function to identify a binary pattern with k ones is a threshold function with unit magnitude weights and threshold $(k - e - 0.5)$, where e is the specified error tolerance. ∎

Note that the 0.5 term in the threshold expression in Theorem 1 makes the threshold comparison more robust. By including this term in the expression, we ensure that the threshold is exactly in the middle of the maximum weighted sum of inputs when the output is 0 and the minimum weighed sum when the output is 1.

To illustrate the theorem, consider an 8 input threshold function $f(x_1, x_2, \ldots, x_8)$ that detects the pattern 10011011 in the presence of up to 2 errors. The threshold function for this uses weights 1 for all inputs that correspond to ones in the target pattern and weights -1 for those corresponding to zeros. Further, the threshold, as specified in the theorem, works out to be $5-2-0.5 = 2.5$. The resultant function is shown in fig. 2

3 NANOTECHNOLOGY IMPLEMENTATION

Integrated circuit designers are constantly thriving to reduce the size of the transistors so as to reduce the size and power, and increase the speed and yield. However, there exists a fundamental size limit beyond which one cannot shrink the transistor. If one goes from the μm scale to nanometer scale, current sub μm Si-CMOS transistors cease to operate. In particular, at base region widths comparable to the electron wavelengths, the potential barrier at base leaks severely, allowing electrons to tunnel from the emitter to the collector. At these sizes, conventional transistors lose their switching ability. On the other hand, this same tunneling effect can be exploited in nanotechnology devices such as resonant tunneling diodes (RTDs) and resonant tunneling transistors (RTTs).

RTDs and RTTs use tunneling of electrons at discrete energy levels in a double barrier quantum well structure. The tunneling phenomena creates the effect of negative resistance giving the current-voltage $(I - V)$ characteristics shown in Fig. 3. RTDs and RTTs are attractive, because they are the most mature of all the nanoelectronic devices [7] and have already been demonstrated and studied by many researchers.

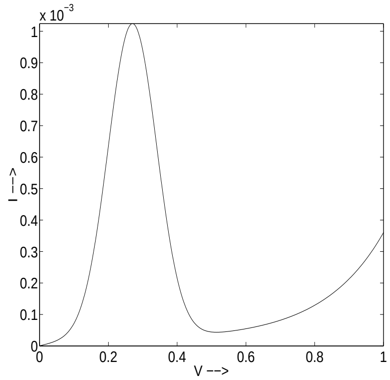

Figure 3: $I - V$ characteristics of a RTD.

Implementation of threshold logic using RTDs uses the *monostable-bistable transition logic element* (MOBILE) [8] shown in Fig. 4. Behavior of the MOBILE is dependent upon the peak currents I_A and I_B of the two RTDs (which depend on the area of the RTD junctions) and the voltage where the current peaks, V_p. As V_{SWP} is increased from 0 to more than $2V_p$ volts, V_{out} settles to a low value (corresponding to a logic 0) if $I_A < I_B$ or a high value (corresponding to a logic 1) if $I_A > I_B$. When the output settles to either of the two values, the current flowing through the circuit is very small.

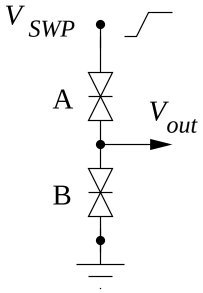

Figure 4: Schematic representation of a MOBILE.

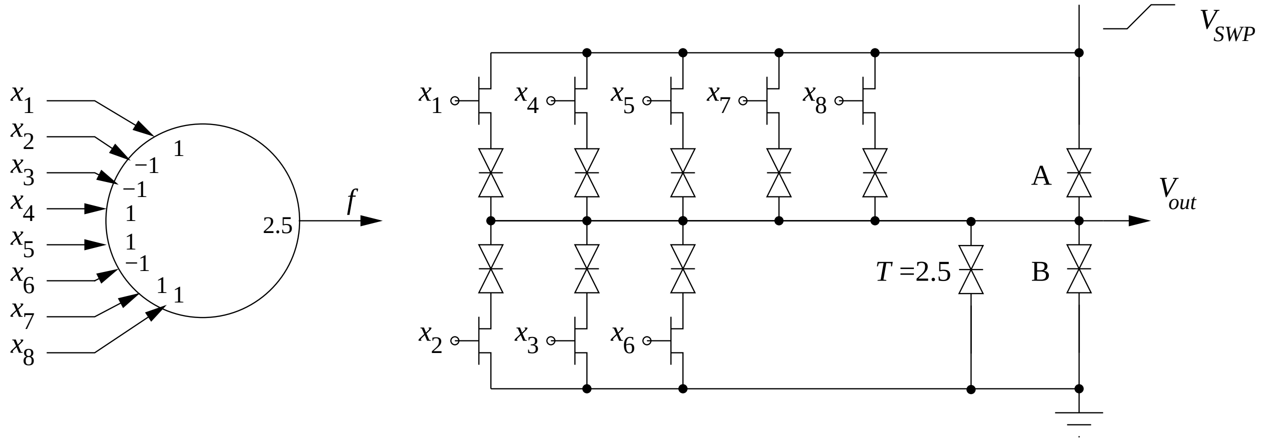

Figure 2: A threshold circuit and its RTD/HFET implementation to detect pattern 10011011 in the presence of up to 2 errors. Note that the area of the threshold RTD is 2.5 times that of all other RTDs.

By connecting multiple RTDs in parallel, one can create an effective peak current which is equal to the sum of individual peak currents. Each individual peak current may be varied by using RTDs of varying areas. In addition, it may be switched in or out of the sum by employing an *heterostructure field effect transistor* (HFET) switch in series[1]. Positively weighed inputs may be placed on top (in parallel with the RTD A) and negatively weighed, at the bottom in parallel with RTD B. Thus the MOBILE architecture allows one to create a weighed sum of the inputs. The comparison of two effective peak currents allows one to compare the weighed sum with a preset threshold. Fig. 2 shows the RTD/HFET implementation of the threshold function used to detect pattern 10011011. Note that because all the input weights are ±1 (Theorem 1), RTDs corresponding to each input have the same area. This minimum area will be called the unit area. To reduce the number of RTDs, the RTD implementing the threshold and the RTD B may be replaced by a single RTD of area 3.5.

The circuit in Fig. 2 was simulated and verified in MATLAB and PSPICE using RTD models in [10].

4 CONFIGURABLE APPROXIMATE PATTERN MATCHING

We now modify the architecture of Fig. 2 to the one shown in Fig. 5 which will be suitable for any target pattern and arbitrary error tolerance. In this new architecture, we allocate two RTD/HFETs corresponding to each bit of the pattern, one at the top and one at the bottom in order to provide for the bit to be either 0 or 1. Since all weights are ±1, all the RTDs used here have unit area. Clearly, only one RTD/HFET of the two allocated will be used for each input bit. We use all the unused RTD/HFETs to set up the threshold to any required value. Assume that the target pattern has k ones. Then the k positive variables will be applied to the top RTD/HFETs and the $n-k$ negative variables at the bottom. Thus $n-k$ and k input terminals are unused at the top and the bottom respectively. Now recall from theorem 1 that for $0 \leq e \leq n$, the threshold is bound by $-(n-k) - 0.5 \leq T \leq k - 0.5$. As shown in Fig. 2, positive thresholds are located at the bottom and negative at the top. Thus the 0.5 portion of the threshold T can be implemented through a RTD of that area at the top as shown in Fig. 5. A negative threshold value bounded in range $-(n-k) \leq T < 0$ can be implemented through the $n-k$ unused RTDs at the top and a positive threshold value in the range $0 < T \leq k$ can be similarly implemented through the k unused RTDs at the bottom. Thus, even though the threshold can take $2n$ possible values from $-n$ (when $k = 0$ and $e = n$) to $+n$ (when $k = n$ and $e = 0$), our architecture allows us to implement it through only n unused RTDs of unit area.

We now describe configuration procedure of the architecture in Fig. 5 for a specified target pattern with k ones and error tolerance of e.

1. Apply positive variables at the top and negative at the bottom.

2. If $e \leq k$, apply 0 to e unused inputs at the bottom and 1 to the rest. Apply 0 at all unused top inputs.

3. If $e > k$, apply 0 to all unused bottom inputs. Apply 1 to $e-k$ unused inputs at the top and 0 to the rest.

Table 1 illustrates matching 10011011, an 8 bit pattern with a large number of patterns $x_1x_2x_3x_4x_5x_6x_7x_8$ with an arbitrary error tolerance e.

[1] The monolithic integration of an HFET with an RTD is known in literature as the three terminal hybrid RTT [9]

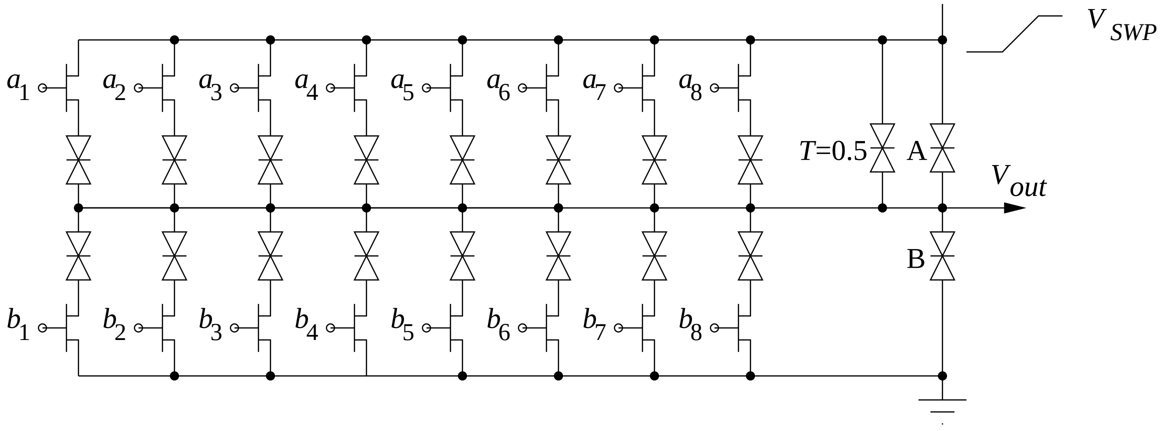

Figure 5: A configurable pattern matching circuit to match two 8-bit patterns with any error tolerance from 0 to 8. Note that the threshold RTD has half the area as that of all other RTDs and may be combined with RTD A.

Table 1: Illustration of configurations of the circuit in Fig. 5.

pattern	e	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
10011011	0	x_1	0	0	x_4	x_5	0	x_7	x_8	1	x_2	x_3	1	1	x_6	1	1
10011011	1	x_1	0	0	x_4	x_5	0	x_7	x_8	0	x_2	x_3	1	1	x_6	1	1
10011011	3	x_1	0	0	x_4	x_5	0	x_7	x_8	0	x_2	x_3	0	0	x_6	1	1
10011011	6	x_1	1	0	x_4	x_5	0	x_7	x_8	0	x_2	x_3	0	0	x_6	0	0

5 CONCLUSION

This paper provides a new nanotechnology architecture for approximate pattern matching. It uses a fixed configuration of $2n+1$ RTDs of unit area and one RTD with 1.5 times the unit area. It can be configured for any n bit target pattern and for any error tolerance between 0 and n. Previous nanotechnology applications have mostly relied upon translating Boolean solutions to threshold gates. This paper demonstrates a new design philosophy of obtaining solutions directly in terms of threshold logic that is well suited for the nanotechnology implementation.

REFERENCES

[1] S. Pramanik and C. T. King, "A hardware pattern matching algorithm on a dataflow," *Computer Journal*, vol. 28, pp. 264–269, March 1985.

[2] R. Sastry, N. Ranganathan, and K. Remedios, "CASM: A VLSI chip for approximate string matching," *IEEE Trans. on Pattern Analysis and Machine Intell.*, vol. 17, pp. 824–830, Aug 1995.

[3] A. N. Du, B. X. Fang, X. C. Yun, M. Z. Hu, and X. R. Zheng, "Comparison of string matching algorithms: An aid to information content security," in *Proc. of the 2nd Int. Conf. on Machine Learning and Cybernetics*, (Xian), pp. 2996–3001, Nov 2003.

[4] M. Giraud and D. Lavenier, "Weighted finite automata in hardware for approximate pattern matching," tech. rep., EDAA Ph.D Forum, DATE '04, France, Feb. 2004.

[5] V. Lohweg, C. Diederichs, and D. Müller, "Algorithms for hardware-based pattern recognition," *EURASIP J. for Applied Sig. Proc.*, vol. 12, pp. 1912–1920, 2004.

[6] S. Muroga, *Threshold logic and its applications.* New York: Wiley-Interscience, 1971.

[7] K. Goser and C. Pacha, "System and circuit aspects of nanoelectronics," in *24th European Solid-State Circuits Conf.*, (The Hague, NL), pp. 18–29, Sep. 1998.

[8] T. Akeyoshi, K. Maezawa, and T. Mizutani, "Weighted sum threshold logic operation of MOBILE (monostable-bistable transition logic element) using resonant-tunneling transistors," *IEEE Elec. Dev. Lett.*, vol. 14, pp. 475–477, Oct. 1993.

[9] C. Pacha, K. Goser, A. Brennemann, and W. Prost, "A threshold logic full adder based on resonant tunneling transistors," *24th European Solid-State Circuits Conf.*, pp. 428–431, 1998.

[10] "Definition of software interfaces." ANSWERS Tech Report (Autonomous Nanoelectronic Systems With Extended Replication and Signalling), Jan. 1999.