Power efficient integer multiplication overflow detection circuit using single electron device based threshold logic gates

Anup Kumar Biswas[1], Nurul Islam[2], Souvik Sarkar[3], Sandip Gharami[4] and #Samir Kumar Sarkar[5]

Dept. of CSE, Jalpaiguri Govt. Engineering College, Jalpaiguri-735102

[2]Ramkrishna Mission Residential College, Narendrapur, West Bengal

[3]Dept. of CSE, Govt. College of Engineering and Ceramic Technology,

[4]Birla Institute of Technology, Kolkata, India

[5] Dept. of Electronics and Telecommunication Engineering, Jadavpur University,Kolkata-700032

su_sarkar@hotmail.com

Abstract

Low power consumption, high operating speed and high integration density equipment(s) are financially indispensable in modern Electronics. Single Electron Device (SED) is one such equipment. Single Electron Devices are capable of controlling the transport of an electron or a few number of electron. A single electron is sufficient to store information in the SED. This paper presents the approach for designing multipliers by using single-electron based device. Multipliers with overflow detection based on serial and parallel prefix computation algorithm are elaborately discussed analytically and designed. The overflow detection circuits works in parallel with a simplified multiplier to reduce the overall area and to increase the speed compared to the classical digital circuits.

1. INTRODUCTION

Single electron based logic gates have already been constructed with clock pulses of 1ns each [1]. The technique of tunneling of an electron is utilized for those gates. This technique may also be used for more complex logical circuit like a case of a binary multipliers and overflow detection [2]. When the number of bits that do not fit into space available then overflow occurs. For example, when an arithmetic logic operation creates a result outside of the range of the representable number, overflow occurs. If overflow occurs, an error flag is generated to indicate "out of range". Well-known that, multiplication for two n-bit integers produces a n+n=2n product. But some electronic architecture only return n-least significant bit out of the 2n-bits and overflow sets in the product cannot be represented correctly with only n bits. For example, IBM's Power Microprocessor family supports a 32-bit by 32-bit two's complements multiply instruction which returns the least significant 32-bits of the 64-bit product and an overflow flag [3].

2. COULOMB BLOCADE AND SINGLE ELECTRON TRANSISTOR

A tunnel junction shown in Fig-1 is considered to be a thin insulating barrier between the two conducting electrodes. If we apply bias voltage, there will be a current flow. Avoiding additional effects, according to first-order–approximation-tunneling current is proportional to the applied bias voltage. In electrical terms, a tunnel junction behaves like a resistor of a constant value depending experimentally upon the barrier thickness. For the discrete nature of electric charge, current following through a tunnel junction is a series of events in which only one electron goes through the tunnel junction. As the electron tunnels the junction, the tunnel capacitance is charged with an elementary charge building up a voltage V=e/C; C=junction capacitance. If the capacitance of the tunnel junction is very small, the voltage developed in the tunnel junction may be adequate to prevent another electron to tunnel. The electrical current is suppressed then for the bias voltage lower than the voltage developed in the tunnel junction and the resistance of the device no longer remains constant. The increment of the differential resistance of the tuunel junction around zero bias is considered as the Coloumb blockade. So, we can define the Coloumb blockade as increased resistance at very low bias voltages of an electronic device which is having at least one low capacitance tunnel junction.

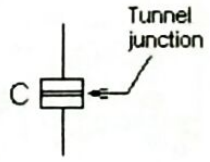

Fig. 1: Tunnel Junction

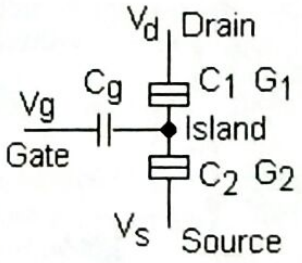

Fig. 2: Single electron Transistor (SET)

The fundamental principle of single-electronics is based on the Coulomb blockade. Single electron tunneling circuits seem to be a promising candidate for future VLSI for its ultra-low power consumption, ultra small size and rich functionality

Single electron Transistor (SET) is shown in Fig-2. A SET has two tunnel junctions having capacitances and conductances C_1 ,C_2 and G_1 ,G_2 respectively ,and share one common electrode with a low capacitance known as **island.** The electric potential of the island can be tuned by a third electrode, called gate, which is capacitively coupled(gate capacitance C_g) to the island. The drain , source and gate voltages are V_d, V_s and V_g respectively . For proper operations of SET both of the conductances G1 and G2 , of course, are to be smaller than 1/Rq; where $Rq = h/e^2 \approx 25.8$ KΩ and charging energy $E_C=e^2/(2C)$[where $C = C_1+C_2+C_g$] has to be greater than thermal fluctuations kT i.e., $E_C>kT$

3. SOME SINGLE-ELECTRONIC THRESHOLD LOGIC GATES

Logic gates can be realized using SETs. In the adjoining Figs. (Fig. 3 and Fig. 4), two input AND gate and two input OR gate are shown realized with single electron device based threshold logic gares. These gates are to be used in our subsequent design and implementation of our proposed circuit.

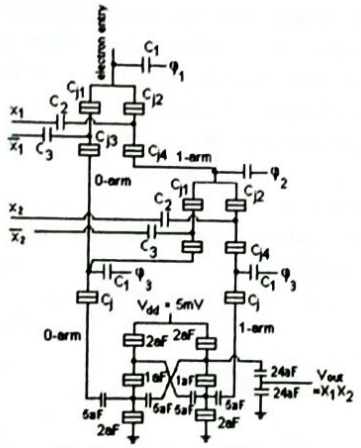

Fig. 3: Two-input AND gate

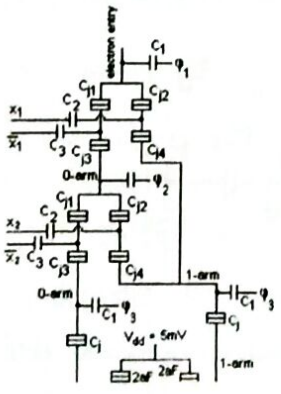

Fig. 4: Two-input OR gate

4. THEOREM RELATED TO OVERFLOW

4.1 Theorem-1

For two unsigned binary codewords having r and m significant bits respectively, no overflow for the product (multiplication) of these two codewords will happen iff total number of significant bit of the product is n≥l+m.

Proof: Suppose two unsigned binary word A and B. Significant bits of A include the leftmost 1 and all the remaining bits right to that. If A=00100010, then A has six significant bits. Similarly if B=10010001, then it has eight significant bits. Now M=A×B, then there will be no overflow if the significant bit of the product "M" is l+m >n i.e., we have to prove for non-overflow r+m>n.

Limit of the value of A is

$2^{r-1} \le A \le 2^r-1$ ..... (i)

Similarly limit of B is

$2^{m-1} \le A \le 2^m-1$ .....(ii)

Limit of the value of M=A×B is

$2^{r+m-2} \le A \le 2^{r+m}-(2^r+2^m)+1$ .....(iii)

from the left hand side of the equation if $2^n \le 2^{r+m-2}$ then overflow occurs. i.e., if

$r+m \ge n+2$.....(iv)

overflow occurs, overflow does not occur if

$r+m<n+2$.....(v)

From the right hand side of the equation (iii) there will be no overflow if

$2^{r+m}-(2^r+2^m)+1 \le 2^n-1$.....(vi)

but any positive integer $-(2^r+2^m) \le -2$

or

$(2^r+2^m)+1 \le -1$.....(vii)

combining equation(vi) and (vii) we get

$2^{r+m} \le 2^n$ or $r+m \le n$.....(viii)

from equations (v) and (viii) we can conclude that no overflow occurs if $r+m \le n$ or if $n \ge r+m$

[proved]

4.2 *Detection of overflow*

From the equation (iv) it is obvious that if $l+m \geq n+2$ then overflow must happen. This condition is satisfied with the equation [4] given below.

$O_{vl}=a_{n-1}.b_1+(a_{n-1}+a_{n-2}).b_2+(a_{n-1}+a_{n-2}+a_{n-3}).b_3+\ldots\ldots+(a_{n-1}+a_{n-2}+\ldots+a_1).b_{n-1}$ ....(ix)

For verification we take first term $a_{n-1}.b_1$, if its value is 1 i.e., $a_{n-1}.b_1=1$ then the total number of significant bits is at least n+2 as A has at least n significant bits and B has at least 2 significant bits. In the same way, if the second term $(a_{n-1}+a_{n-2}).b_2$ is 1 then A has at least (n-1) significant bits and B has at least 3 significant bits. Similarly, if the last term $(a_{n-1}+a_{n-2}+\ldots+a_1).b_{n-1}$ is 1, then A has at least 2 significant bits and B has at least n significant bits so the total number of significant bits will be (n+2). The equation (ix) is written as

$O_{vl}=\sum_{i=1}^{n-1} \sum_{j=1}^{n-1} a_{n-j}.b_i$ where bit-dot-product and

bit-summation indicates logical

AND and OR respectively. Comparing equation (iv) and (viii) We get that when $l+m=n+1$ the multiplied value may overflow but the value is less than 2^{n+1}. In this situation the product bits p_0 through p_n are sufficient to indicate whether the product value will overflow or not. If m_n is equal to 1 then only overflow occurs. So we conclude that if $l+m \geq n+2$ or $m_n=1$ then overflow must occur.

But from equation (ix), the essential number of AND and OR gates are respectively

$1+2+3+\ldots+(n-1)=(n-1).n+2$ ....(x)

And

$(n-2)+(1+2+3+4+\ldots+n-2) = \{(n-1).n+2\}-1$.........(xi).

If the recursive (iterative) technique is applied then the number of AND or OR gates can be reduced.

4.3 *Iterative technique*

The hardware reduction can be used here by using the iterative process. We take

$r_{k+1}=r_k+o_{k+1}.b_k$ and $o_{k+1}=o_k+a_{n-k}$

where $2 \leq k \leq n-1$.

Initially we take, $o_{k=2}=a_{n-1}$ and $r_{k=2}=a_{n-1}.b_1$

Then $o_3=o_2+a_{n-2}=a_{n-1}+a_{n-2}$

$r_3=r_2+o_3.b_2=a_{n-1}.b_1+(a_{n-1}+a_{n-2}).b_2$

$o_4=a_{n-1}+a_{n-2}+a_{n-3}$

$r_4=r_3+o_4.b_3$

$= a_{n-1}.b_1+(a_{n-1}+a_{n-2}).b_2+(a_{n-1}+a_{n-2}+a_{n-3}).b_3$

....

$r_n=O_{vl}=a_{n-1}.b_1+(a_{n-1}+a_{n-2}).b_2+(a_{n-1}+a_{n-2}+a_{n-3}).b_3+\ldots+(a_{n-1}+a_{n-2}+\ldots+a_1).b_{n-1}$

.........(xii)

After (n-1) iterations we obtain the result of the equation (xii) which indicates the overflow. if the significant bits of the product M is $\geq n+1$ then also overflow occurs As overflow when $P>2^n$. So we can draw the block diagram for overflow using multiplier and iterative circuit like as

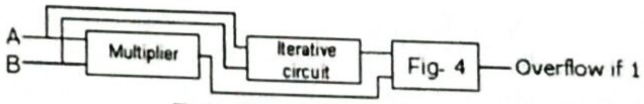

Fig.5: Block diagram for overflow

4.4 *Implementation of multiplier*

From Theorem-1 and from the iteration equation (xii), We can implement an unsigned multiplier and a overflow circuit. First eight bit unsigned digital multiplier has been implemented in Fig. 5.

4.5 *Reduction of delay*

The unsigned digital overflow can be reduced by using the prefix algorithm [5]. The equation (xii) are rewritten as

$O_{vl}=A_{n-1}.b_1+A_{n-2}.b_2+A_{n-3}.b_3+\ldots\ldots+A_1.b_{n-1}$ ......(xiii)

where $\sum_{k=1}^{n-1} a_k$ ....(xiv)

4.5.1 *Serial Prefix*

For a size n, assume that assume that the inputs are d_1, d_2, d_3, d_4, ..., dn, and the related operation used is "*". After applying "*" each output y_i is written as

$y_i=d_1*d_2*d_3*d_4*\ldots*d_{i-1}*d_i$.........(xv), $1 \leq i \leq n$.

Equation (xv) can be utilized for sequentially computing y_i by using (n-1) times of "*" operations. Then this process is called serial prefix computation.

4.5.2 *Parallel prefix*

From equation (xv), it is clear that we can apply the operation "*"in any order. For example, we can determine (d_1*d_2), $(d_3$

$*d_4$),.....or ,(d_{i-1} * d_i) in parallel. This technique is called parallel prefix [6] technique. For reduction of time delay parallel prefix technique can be applied.

$A_k = \sum_{j=1}^{k} a_j$, $1 \le k \le n-1$, can be implemented using parallel prefix technique for an n = 8 bit using the operator "OR". Parallel prefix technique is given in Fig-6 and Fig. 7.

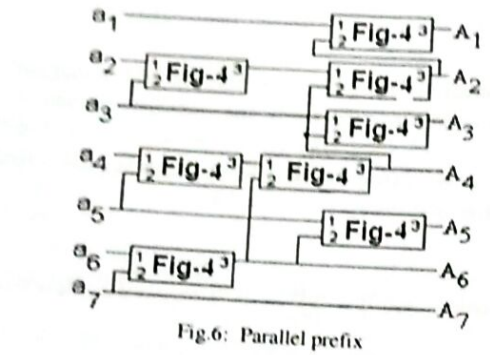

Fig.6: Parallel prefix

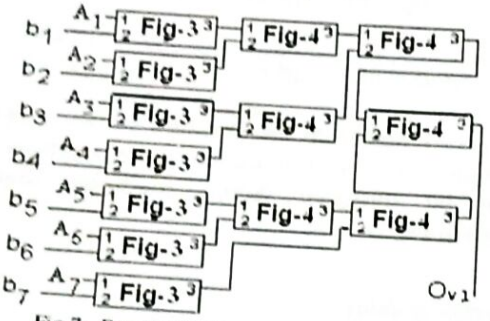

Fig.7: Parallel Prefix technique for overflow circuit for $A_k = \Sigma a_j$ (O_{v1})determinati

5. Results and Discussions

A general technique for multiplication along with the overflow based on single electron tunneling phenomena has been implemented for positive (unsigned) multiplication. If both the operands are of same sign the result of the multiplication will be of positive sign. In such situation the Fig. 7 is used for overflow detection. The unsigned multiplication circuit is brought to a same platform for unifying them after a small modification of the input data. The overflow detection circuit does not depend upon the internal carries which is generated by the multiplier and does not require the partial product of the multiplier to be modified. So the circuit can be applied to any type of simplified multiplier which produces (n+1) product bits. Multiplier circuit is not modified from serial prefix circuit and multipliers are treated as the tree multipliers. Parallel prefix algorithm is used for time saving and decreasing the gates or nodes. The designed multiplier circuit can be used efficiently for unsigned operands. Obviously, Single electron device based circuit is at least 3 times faster than the classical logic based circuits [7]. The designed circuit (simulated) is verified with same trial operands multiplications. The results are found to be satisfactory and the speed of operation is observed to be at least three times faster than the conventional multiplier circuits.

6. Conclusion

A general technique for multiplication along with the overflow based on single electron tunneling phenomena has been implemented for positive (unsigned) multiplication. This technique includes separately serial and parallel prefix algorithm. Parallel prefix algorithm is used for time saving and decreasing the gates or nodes. Tunneling, cascading and pipelining are happened in these circuits. Single electronics has the potency to operate with very low supply power and quantum properties that appear a nanometer scale in pipeline represent for thinking only a single electron tunneling through the tunnel junction(s). According to Heisenberg's uncertainty principle, we can predict that speed power product of an electron lie close to the quantum limit. So electron(s) can move through the single electron circuits with nearly electronic speed. The speed of operation of the present multiplier is found to be much faster than conventional multiplier circuits [7]. Thermal agitation and stochastic errors caused by probabilistic fluctuations are excluded in our design.

Acknowledgement

Subir Kumar Sarkar thankfully acknowledges the financial support obtained from DRDO, Govt. of India (ERIP/ER/0503561/M/01/905 dt.01/08/2006)

References

[1] N. Asahi et al.: "Single electron logic systems based on BDD" IEICE Trans. ED. E-81-C Jan. 1998.

[2] Mustafa Gok, M. J. Schulte and M. G. Arnold: Integer multipliers with overflow detection, IEEE Transaction on Computers 55, 8, 1062-1066, 2006.

[3] IBM, Power PC Microprocessor Family: The Programming Environments for32-bit Microprocessors, no. G522-0290-01, Feb.2000.

[4] M. J. Schulte, et al. "Integer Multiplication with Overflow Detection or saturation" IEEE Trans. Computers, vol.49, no. 7, pp. 681-691, July 2000.

[5] Y. H. Cha, G.Y. Cho, H.H.Choi and H.B.Song,"Bit result Integer Multiplier with overflow detector," IEE Electronic Letters, voJuly 2001.

[6] R. E. Ladver and M.J. Fiscer, "Parallel Prefix Computation" J. ACM, vol. 27, pp. 831—938, Oct. 1980.

[7] A. K. Biswas and S. K. Sarkar: "An Arithmetic logic unit of a computer based on single electron transport system", Semiconductor Physics Quantum Electronics and opt electronics. Vol. 6, No.1 pp.91-96, 2003.