International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 5 (2019), pp. 679-685
© International Research Publication House. http://www.irphouse.com

A VLSI Design of a High-Speed Parallel Digital *k*-Winners-Take-All Circuit

Myungchul Yoon
Professor, Department of Electronics and Electrical Engineering,
Dankook University, Yongin, Gyeonggi-do, Republic of Korea

ORCID: 0000-0001-7952-4349

Abstract

The *k*-Winners-Take-All (*k*WTA) is an operation to find the largest *k*-input (*k*>1) among *N* inputs. Unlike analog *k*WTA circuits, it was very difficult for digital *k*WTA to search all k-winners simultaneously. A high-speed parallel digital *k*WTA circuit called HDkW which compares all inputs simultaneously is presented in this paper. To achieve a high-speed operation, the HDkW employs some speed enhancement techniques such as pre-charging, block-bypassing, and fast termination to the NMOS demultiplexer array circuit (M-chain). The effect of these schemes on the speed is verified quantitatively by simulations, and the speed of the HDkW with each scheme is compared with that of another existing parallel digital *k*WTA circuit (PDk). The speed increase varies widely according to the number of inputs, the size of *k*, the level of M-chains, and block-partitioning methods. Generally, the HDkW get the higher speed enhancements for searching the fewer number of winners, and for searching in the larger number of inputs. The pre-charged M-chain makes the speed of the HDkW about 1.45 times faster than that of the PDk, and the block-bypassing scheme with fast termination scheme does 1.4~12 times faster than that of the previous PDk. Overall, the HDkW is 1.9~17.5 times faster than PDk

Keywords: Digital *k*-WTA circuit, *k*-Winners-Take-All circuit, Parallel *k*-WTA, Parallel search, Scalable *k*-WTA architecture.

I. INTRODUCTION

The Winner-Take-All (WTA) is an operation to search for the largest (or smallest) input among *N* inputs, and the *k*-WTA is an extension of WTA such that it searches the largest (or smallest) *k*-input among *N* inputs. The *k*WTA operation is used in many areas that need to select *k*-input which meets the given conditions most closely. For example, it is used in machine learning [1], neural networks [2], image processing [3], signal processing [4], pattern recognition [5], filtering [6], vision systems [7], clustering [8], sorting [9], and information retrieval [10], etc

Many applications use analog *k*WTA circuits [11][12][13] because an analog *k*WTA circuit can be implemented in much smaller area than a digital *k*WTA circuit and it is possible to search all *k* winners in parallel. However, analog *k*WTA circuits have many disadvantages compared to digital kWTA circuits. Unlike digital circuits, analog *k*WTA circuits suffer from matching problem [14] and stability-convergence problem [15], and they have difficulties in achieving a high-precision operation. On the contrary, digital *k*WTA circuits are free from the mismatch and convergence problems, and easy to control their precision.

Although many areas still use analog inputs and *k*WTA circuits, the needs for using digital circuits and corresponding digital *k*WTA circuits grow rapidly. One of the biggest obstacles for digital *k*WTA circuits was to find a method to compare all inputs simultaneously to search the *k*-winner. The design of an efficient parallel digital *k*WTA circuit was a difficult problem, even though some parallel search architectures [16] [17] were developed for the digital WTA operation. A hardware intensive parallel *k*WTA architecture was proposed in [18], but its $O(N^3)$ hardware complexity limits its usage only for a small number of digital inputs. Most of digital *k*WTA circuits either iterate the WTA operation k-times [19], or compare inputs serially with a pipelined architecture [20].

Recently, a parallel-search algorithm for digital *k*WTA and its VLSI implementation were proposed in [21]. The circuit presented in [21] compares all inputs simultaneously to find *k*-winners so that it is quite useful for the applications with many inputs. The circuit shows that the parallel *k*WTA operation is possible for digital inputs as well.

In this paper, a high-speed parallel digital *k*WTA circuit called HDkW (High-speed Digital *k*WTA circuit) is presented. The HDkW circuit enhances the circuit in [21] to increase the speed of *k*WTA operation greatly. HDkW is a general purpose parallel digital *k*WTA circuit which can be employed into any application that needs a fast digital *k*WTA operation.

The structure and operation of the HDkW circuit are described in Section II, while the schemes developed for speed enhancement are presented in Section III. The effects of the newly developed schemes are estimated by simulations and the results are described in in Section IV, and Section V concludes this paper.

II. THE STRUCTURE AND OPERATION OF HDkW CIRCUIT

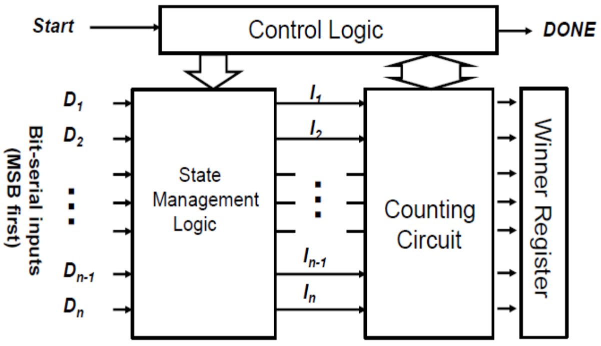


Fig. 1. Block diagram of the HDkW circuit

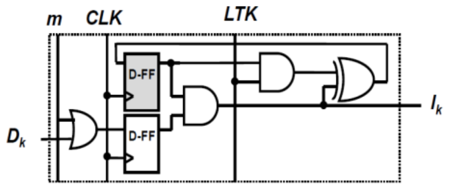


Fig. 2. Logic diagram for a cell of the state management logic

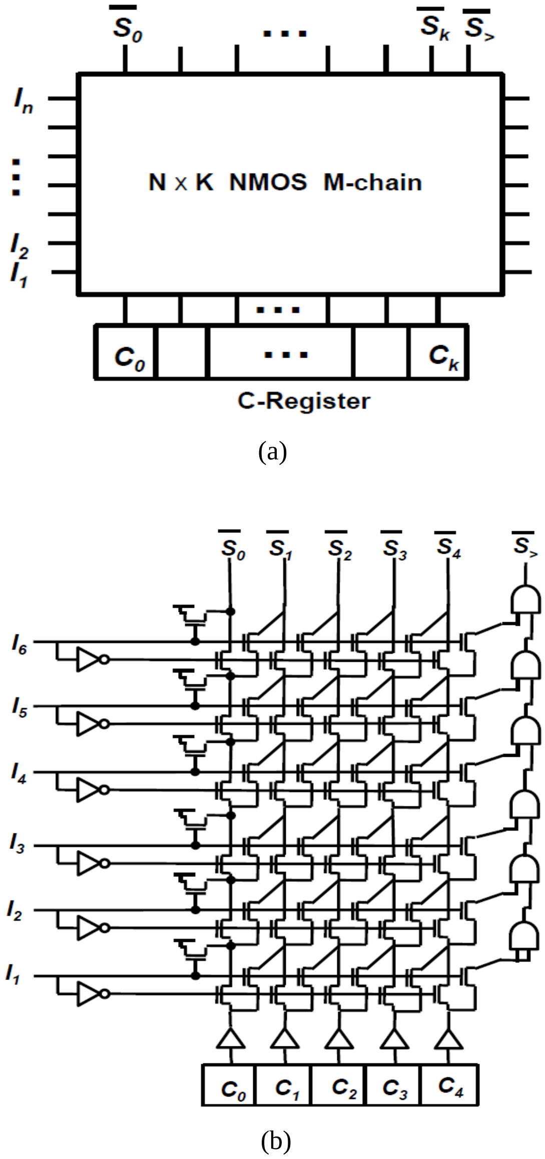
HDkW is developed to increase the speed of the parallel digital *k*WTA (PDk) circuit presented in [21]. The skeleton of HDkW circuit is similar to that of PDk. The block-diagram of HDkW is shown in Fig. 1. HDkW consists of two major parts: the state-management-logic and the counting-circuit. The state-management-logic updates the state of inputs, and the counting-circuit is used to obtain the number of winners.

Inputs can have one of three states, Confirmed-Winner, Confirmed-Loser and Competitor. A Competitor can be either of the two temporal sub-states: Instant-Winner and Instant-Loser. The state-management-logic is composed of *N* identical cells (one cell per input) which operate independently in parallel. The logic diagram of the cell is depicted in Fig. 2. In each cycle, one bit per input, D_k , is fed to the cell. The state of an input is stored in the shaded flip-flip (Fig. 2). If 1 is stored in the FF, the input is in Competitor state. If the output of the FF is 0, it is either Confirmed-Winner or Confirmed-Loser. Confirmed-winners are stored in the Winner-Register in Fig. 1, and do not join in competition any more. Therefore, only competitors are joined in competition to be a confirmed winner.

The operation of the HDkW is performed in a sequence of cycles. To find the *k* biggest (or smallest) numbers from *N* of *m*-bit inputs, the HDkW simultaneously compares all inputs, one bit a cycle from MSB to LSB. *k*WTA operation is initiated by the Start signal (Fig.1), and the DONE signal is generated to inform the end of operation. If all inputs have distinct values, *k*-winners can be found within *m* cycles. In most cases, it takes less cycles, because the HDkW finishes the operation as soon as *k*-winners are identified. If more than two inputs with the same magnitude become the *k*-th winner, one more cycle is required for tie-break. The signal *m* in Fig. 2 becomes 1 only in the tie-break cycle.

The *k*WTA operation of the HDkW circuit can be described as follows. At the beginning all inputs are set to competitor state. After that, the *k*WTA operation of the HDkW circuit is the repetition of the following operations in each cycle until DONE is set to 1.

- From MST to LSB, one bit (D_i) per input is fed to the state-management-logic
- Each cell in the state-management-logic generates its instant-winner state (I_i) with the input bit (D_i) and its current input state such that if input-state is Competitor and D_i=1, then it is Instant-Winner so that I_i =1. If input-state is Competitor and D_i=0, it is Instant-Loser so that I_i =0. Assign I_i =0 for all inputs in Confirmed-Winner or Confirmed-Loser state.
- The I_i 's are fed to the counting-circuit.
- With I_i 's as the inputs, the winner-decision circuit counts the number of 1s in I_i 's i.e. the number of instant-winners, and adds it to the number of confirmed winners to obtain the value *S*.
- If $S > k$, make the signal LTK (less than k) to 0, otherwise, make LTK to 1 and store all instant-winners to the Winner-Register.
- If *S*=*k*, then finish *k*WTA operation (DONE=1), else, send LTK signal to to the state-management-logic
- The state-management-logic updates the state of inputs according to LTK signal:
  - if LTK=1, switch all Instant-Winners to Confirmed-Winner, but all Instant-Losers remain as Competitor.
  - if LTK=0, switch all Instant-Losers to Confirmed-Loser, but all Instant-Winners remain as Competitor
- Go to the next cycle.



(a)

(b)

Fig. 3. The structure of the counting-circuit (a) symbolic diagram of the N×k counting circuit (b) example for a 6×4 counting-circuit implemented by NMOS deMUX-chain.

In the above operation, the most critical operation for speed is counting the number of instant-winners in the counting-circuit. All the other operations are performed independently with bit-wise parallelism. Therefore, the design of a fast counting-circuit is the key of the HDkW circuit.

The circuit counting the number of 1s in many bits is designed by using an 1-to-2 demultiplexer (deMUX) chain. Fig. 3 shows the NMOS deMUX-chain (M-chain) circuit which is used to count the total number of winners in each cycle. The $N\times k$ M-chain is composed of $N\times(k+1)$ array of 1-to-2 deMUXs. The C-register in Fig. 3 is used to store the number of confirmed-winners. All bits in the C-register are filled with 1, except only one bit, C_p , where p is the number of confirmed winners. At the beginning, $C_0 = 0$, and the register is updated at the end of each cycle. In Fig. 3-(b) the structure of a 6×4 NMOS M-chain is shown as an example. The $\overline{S_i}$ represents the sum of the number of the confirmed-winners and the instant-winners

The M-chain can find the number total winners (the confirmed winners and the instant winners) by the following method. In each cycle, 0 in the C-register starts its travel to the top. If $I_i = 0$ deMUX passes 0 vertically-upward, while it goes right-upward when $I_i = 1$. By this way, 0 at C_0 reaches $\overline{S_l}$ when the number of 1s in the I_i 's is l, regardless of locations of 1s. If the number of confirmed winners is p and the number of instant winners is l, 0 is at C_p and it is propagated to $\overline{S_{p+l}}$. When $\overline{S_r} = 0$ $(r < k)$, the signal LTK (less than k) becomes 1. If $\overline{S_k} = 0$, the search is finished and the signal DONE becomes 1. $\overline{S_>}$ becomes 0 when the sum S exceeds k.

In Fig. 3-(b), the counting-circuit is constructed by a single-level M-chain. Although all inputs are compared simultaneously, the counting is not a fully parallel operation. For many inputs, a multi-level M-chain[21] as in Fig. 4 can be used to increase its speed by using parallelism. The multi-level M-chain requires more hardware overheads so that it is necessary to trade-off between the speed and the size of the circuit.

III. THE SPEED ENHANCEMENT SCHEMES OF THE HDkW CIRCUIT

As described in Section 2, the speed propagating **0** from the C-register to $\overline{S_x}$ governs the total delay of the HDkW circuit. Therefore, increasing the speed of M-chain is the key of the high-speed *k*WTA. In Fig. 3, the deMUX is designed by two NMOS pass-transistors which is the simplest implementation for a deMUX circuit. As the first approach, several different deMUX implementations are tested for speed enhancement. For example, the pass-transistors are replaced by transfer-gates, inverters, buffers, etc. As the result of simulations, it is concluded that no other deMUX circuits are faster than the pass-transistor implementation. Therefore, the deMUX implemented by two NMOS pass-transistors is the best choice not only for the speed but also for the hardware simplicity. To decrease the delay of the NMOS M-chain, two circuit techniques, pre-charging and bypassing, are employed in the HDkW.

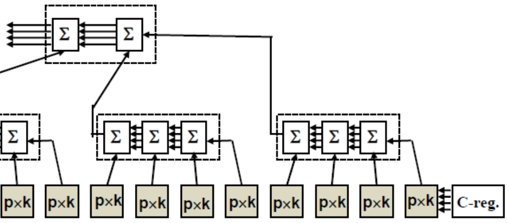


Fig. 4. Block diagram of the multi-level M-chain circuit

III.I PRE-CHARGED NMOS DEMUX-CHAIN

Although the NMOS M-chain structure shown in Fig. 3 is efficient in the speed and size, there is a problem which degrades the speed of the circuit. The delay of M-chain mainly due to the propagation time of 0-signal from $\overline{C_p}$ to $\overline{S_i}$. However, the speed of M-chain depends on the propagation of 1-signal from $\overline{C_l}$ to $\overline{S_j}$, as well. The reason is that the operation of the HDkW is composed of a series of stages. For a correct operation, the values of $\overline{S_i}$'s should be evaluated not only after the full development 0-path in the current stage ($\overline{S_{new}}$), but also after the full restoration of 0-path in the previous stage ($\overline{S_{old}}$). The delay of the M-chain is determined by the slower process. By the property of NMOS transistors, the propagation of 0 in the NMOS chain is faster than that of 1. Therefore, the speed mismatch between developing new-path and restoring the previous-path may degrade the speed of the M-chain circuit. This problem is more serious for VLSI chips of lower V_{DD} because the V_T/ V_{DD} ratio is larger for lower $V_{DD,}$ so that the speed difference between the pull-up of $\overline{S_{old}}$ and the pull-down of $\overline{S_{new}}$ becomes greater.

The pre-charged NMOS M-chain as in Fig. 5 is employed to remove the speed mismatch problem. As shown in Fig. 5, several rows of nodes in the M-chain are modified as the pre-charge nodes. The interval between pre-charge nodes would be determined such that the delay of the pre-charge circuitry may not increase the delay of the entire operation. As soon as one of $\overline{S_i}$ signals is developed, the values of $\overline{S_i}$ are latched and the signal *PreCH* becomes 1. The *PreCH* signal activates the pre-charge circuits to restore all nodes in the M-chain to 1. At the beginning of each stage, pre-charge circuit is disabled by *Eval*. By employing the pre-charge circuit, the 0-path developed in a stage is restored before the beginning of the next stage. Therefore, the speed mismatch problem does not occur in the pre-charged M-chain.

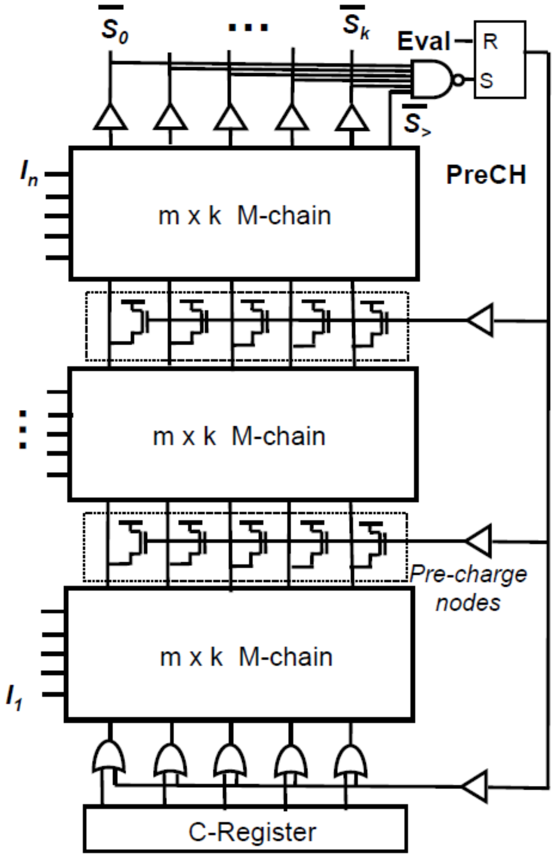


Fig. 5. The structure of the pre-charged NMOS deMUX-chain circuit.

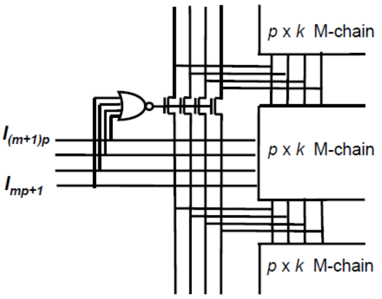
III.II BLOCK-BYPASSING WITH FAST TERMINATION CIRCUIT

The second scheme used to increase the speed of the M-chain consists of two parts, the block-bypassing scheme and the fast termination scheme. The delay of the M-chain can be reduced when these two parts cooperate with each other.

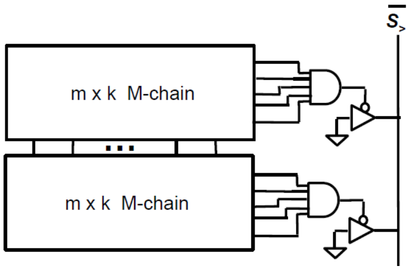
Fig.6-(a) shows structure of the block-bypassing scheme. The $N \times k$ M-chain is divided into q-blocks where each block is composed of $p \times k$ (pq=N) M-chain. A bypassing route is prepared for each block and it is activated when all I_j inputs to this block are 0. If all I_j are 0, all $\overline{C_l}$ inputs of the block jump to the next block through the bypass circuit rather than propagate through the M-chain.

The fast termination (FT) circuit is shown in Fig. 6-(b). A large driver per m inputs is placed to pull down the $\overline{S_>}$ line quickly. It needs not match the number m to p of the bypassing block. The fast termination happens when the sum of the confirmed winners and the instant winners exceed k. In this case, no further counting operation is necessary so that the FT circuit in Fig. 6-(b) pulls down the $\overline{S_>}$ line as quickly as possible.

In the worst case, the block-bypassing or FT circuit alone cannot reduce the delay of the M-chain. To reduce the delay of the M-chain, the block bypassing scheme must be employed with the fast termination circuit. In the early stages of *k*WTA operations, lots of inputs are instant winners so that the effect of the block-bypassing is negligible, but the fast termination circuit is activated frequently. On the other hand, block-bypassing occurs mostly in the late stages of the operation, because the number of instant winners is very small.



(a)



(b)

Fig. 6. The structure of block bypassing and fast termination for a single level M-chain (a) the structure of the block-bypassing scheme (b) the structure of the fast termination scheme.

When both of the schemes are employed, the delay of the M-chain (t_M) is bounded by

$$t_M \le k t_{block} + (q-k) t_{bypass} \tag{1}$$

where t_{block} is the delay of 0-signal to propagate through a block, and t_{bypass} is the delay passing a bypass transistor. It is noted in Eq. (1) that the delay of the M-chain with the bypassing and FT circuit depends on k, while the delay of the original M-chain is almost independent of k.

Applying the bypassing with FT scheme to the multi-level M-chain structure[21] is a little more complicate. The bypassing scheme is applied only in the top level M-chain, because the worst case delay is not affected even though it is applied to other level M-chains. On the other hand, the FT circuit is applied to all levels of M-chains.

IV. EXPERIMENTS

The performance of the HDkW is evaluated by simulations. A series of experiments are performed by SPICE simulation for various numbers of inputs and various structures of the M-chain. The simulation is performed by HSPICE with IBM's "1.2V-0.13μm 8RF-LM" model parameter.

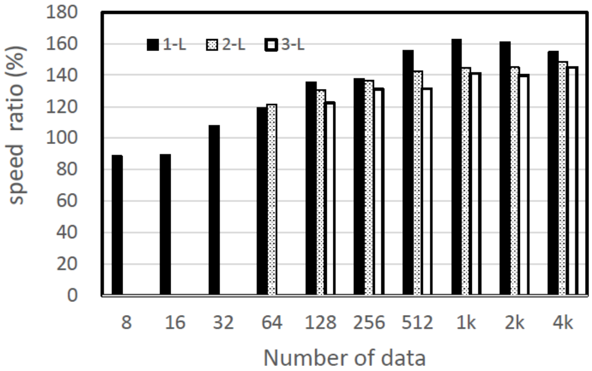
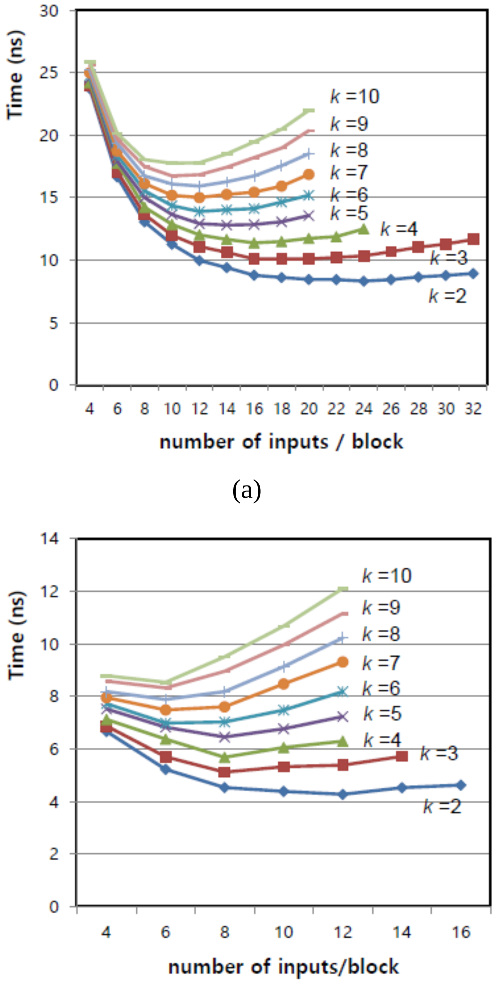


Fig. 7. The speed ratio (t_{ori}/t_{pre}) of the pre-charged M-chain to the original M-chain for various number of inputs

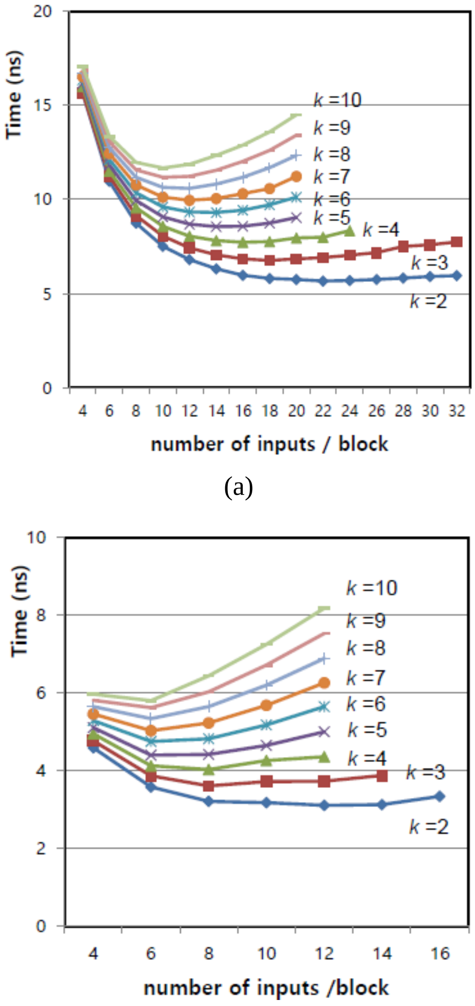


(a)

(b)

Fig. 8. The clock cycle time of the M-chain with the bypassing and FT scheme for various *k* and block sizes (a) N=1024 (b) N=256

The performance of the HDkW is compared with that of other existing parallel digital *k*WTA circuits. However, the *k*WTA circuit (PDk) with the original M-chain [21] is the only existing parallel *k*WTA circuit. The speed comparisons between the PDk and other sequential digital *k*WTA circuits are presented in [21] which indicates that the speed of PDk is much faster than that of other sequential digital *k*WTA circuits. For the reasons, the comparison of HDkW to other parallel digital *k*WTA circuits is focused on speed increase against the PDk only.



(a)

(b)

Fig. 9. The clock cycle time of the M-chain with both of the pre-charge and the block-bypassing schemes for various *k* and block sizes (a) N=1024 (b) N=256

At first, the speed increase of each scheme is estimated. The speed increase of the pre-charged M-chain vs. the original M-chain is evaluated for various inputs with three different structures of M-chain: single level M-chain, 2-level M-chains, and 3-level M-chains. The result of the simulation is shown in Fig. 7. For the small number of data which is less than 32 inputs, the speed of *k*WTA operation with the pre-charged M-chain is degraded because the delay overheads of pre-charge circuitry overweigh its speed gain. For the inputs greater than 32, the pre-charged M-chain is faster than the original one. The speed increase of the pre-charged M-chain is more effective for the longer chain. Although the delay of the M-chain shows a big difference between the single level M-chain and hierarchical multi-level M-chains, the speed enhancement by the pre-charge

Table 1. Speed ratio of HDkW with new schemes vs. PDk (t_{PDki}/t_{HDkW}) for the single level M-chain

		Number of inputs				
		64	**128**	**256**	**512**	**1024**
pre-charge only		1.19	1.36	1.38	1.56	1.63
bypass only	k=2	2.55	3.78	5.42	8.12	11.92
	k=3	2.32	3.04	4.53	6.80	9.83
	k=4	2.14	2.83	4.07	5.92	8.71
	k=5	1.86	2.61	3.59	5.30	7.74
	k=6	1.77	2.48	3.32	4.91	7.14
	k=7	1.65	2.36	3.09	4.64	6.60
	k=8	1.56	2.27	2.94	4.39	6.23
	k=9	1.41	2.10	2.78	4.11	5.92
	k=10	1.34	2.03	2.71	3.91	5.58
precharge + bypass	k=2	3.50	5.16	7.44	12.01	17.50
	k=3	3.05	4.31	6.41	9.46	14.65
	k=4	2.78	4.14	5.74	8.39	12.83
	k=5	2.60	3.71	5.26	7.61	11.59
	k=6	2.60	3.51	4.87	7.12	10.65
	k=7	2.26	3.35	4.60	6.88	9.96
	k=8	2.12	3.18	4.33	6.26	9.36
	k=9	2.01	3.06	4.12	5.97	8.87
	k=10	1.90	2.92	3.99	5.83	8.50

scheme shows little difference between levels. As in Fig. 7, the HDkW with the pre-charged M-chain is up to1.63 times faster than that of PDk with a single level M-chain and about 1.45 times faster with 2-level (2-L) and 3-level (3-L) M-chains.

The speed improvement by the block-bypassing with FT circuit is simulated for various block partitions and different numbers of *k*. Fig. 8 shows the minimum clock cycle time of HDkW with bypassing scheme simulated for 1024 inputs and 256 inputs. As expected by Eq. (1), the M-chain with the bypassing scheme is faster for smaller *k* even for the same number of inputs. Its speed also depends on the number of inputs per block (*p*). If we assume that the delay of the NMOS chain is proportional to the length of the chain, t_{block} can be expressed as t_{bypass} such that

$$t_{block} = \lambda p t_{bypass} \qquad (2)$$

where λ is ratio of the propagation delay of one NMOS transistor in the block to that of bypass transistor. By simple calculation with Eq. (1) and Eq. (2), the delay t_M becomes minimum when *p* is

$$p \approx \sqrt{N / \lambda k} \qquad (3)$$

The minimum cycle time of each curve in Fig. 8 happens at the *p* near the value given by Eq. (3) (let λ=1).

Fig. 9 is obtained by similar simulations for the HPkW employing both of the pre-charge scheme and the block-bypassing scheme. The pattern of curves are almost the same with Fig. 8, but the minimum cycle time is much shorter than that in Fig. 8.

The speed improvements of the HPkW over the PDk are given in Table 1 and Table 2. When a single level M-chain is used, the HPkW is about 2~17 times faster than the PDk depending on the number of *k* and inputs. The greater speed enhancements can be obtained when the larger number of inputs are used and when the smaller number of *k* is searched.

Table 2. Speed ratio of HDkW vs. PDk (t_{PDki}/t_{HDkW}) for 2-level M-chain

	Number of inputs				
	256	**512**	**1024**	**2048**	**4096**
k=2	3.45	4.22	4.56	5.52	8.54
k=3	2.88	3.57	3.93	4.69	7.31
k=4	2.59	3.26	3.60	4.30	6.63
k=5	2.41	3.01	3.36	4.01	6.22
k=6	2.27	2.78	3.16	3.78	5.92
k=7	1.92	2.45	2.76	3.31	5.20
k=8	1.80	2.31	2.53	3.10	4.97
k=9	1.69	2.21	2.42	2.99	4.73
k=10	1.57	2.10	2.33	2.83	4.52

The speed improvement of the HPkW with multi-level hierarchical M-chain is much complicated. The speed of the HPkW depends on the number of *k* as well as how to build blocks in each level. It is difficult to find the best combination of block partitions on each level. For the multi-level M-chain HPkW, the pre-charge scheme is applied to all levels but the block-bypassing scheme is employed to the top-level only. For the 2-level M-chain HPkW, the cycle time of the HPkW has the minimum when *r* and *p* are chosen such that $r \approx \sqrt[3]{kN}$ and $p \approx r / k$, where *r* is the number of inputs in the bottom level block, and *p* is the number of Σ-circuits per bypassing block on the top level. The simulation results for the speed improvement of HPkW vs. PDk with the 2-level M-chain are given in Table 2. Compared with Table 1, the speed improvements for the multi-level M-chain are smaller than those for a single level M-chain because the total length of signal propagation is much shorter in the multi-level M-chains. Nonetheless, the HPkW with the 2-level M-chain is over 2 times faster than the PDk in most cases.

V. CONCLUSION

A high-speed digital *k*WTA circuit (HDkW) is presented in this paper. The high-speed operation is achieved by reducing the delay of NMOS deMUX array (M-chain) by employing the pre-charged M-chain scheme, the block-bypassing and the fast termination schemes. The simulation results show that the HDkW becomes 2~17 times faster than the PDk by employing the new schemes. The speed enhancement of the HDkW is greater when applying larger number of inputs, using lower level M-chains, and searching smaller *k*-winners. The HDkW with a single level M-chain is 17.5 times faster than the PDk when searching 2-winners among 1024 inputs while it is 8.5 times faster for 10-winners. With the 2-level M-chain, it is 4.6 times faster for 2-winners and 2.3 times faster for 10-winners. The great improvement of speed proves the effectiveness of the new schemes.

REFERENCES

[1] C. A. Marinov and J. J. Hopfield, 2005, "Stable computational dynamics for a class of circuits with O(N) interconnections capable of KWTA and rank extractions". IEEE Trans. Circuit. Syst., vol.52, no.5, pp. 949–959.

[2] M. Rahman, K. L. Baishnab, and F. A. Talukdar, 2009, "A high speed and high resolution VLSI Winner-take-all circuit for neural networks and fuzzy systems" IEEE ISSCC2009, pp. 1-4.

[3] A. Fish, D. Akselrod, and O. Yadid-Pecht, 2004, "High precision image centroid computation via an adaptive k-winner-take-all circuit in conjunction with a dynamic element matching algorithm for star tracking applications". Analog Integ. Circuit. Signal Process. vol. 39, pp. 251–266.

[4] A. K. J. Hertz and R. G. Palmer, 1991, Introduction to the Theory of Neural Computation, Redwood City, Addison-Wesley.

[5] D. Tian, Y. Liu, and D. Wei, 2006, "A Dynamic Growing Neural Network for Supervised or Unsupervised Learning," Intelligent Control and Automation, WCICA 2006, vol.1, pp. 2886-2890

[6] U. Cilingiroglu and T. L. E. Dake. 2002, "Rank-order filter design with a sampled-analog multiple-winners-take-all core," IEEE J. Solid-State Circuits, vol. 37, no. 2, pp. 978-984, Aug.

[7] L. Itti, C. Koch, and E. Niebur, 1998, "A model of saliency-based visual attention for rapid scene analysis," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 11, pp. 1254 – 1259.

[8] T. Eltoft and R. I. P. deFigueiredo, 1998, "A new neural network for cluster-detection-and-labeling," IEEE Trans. Neural Networks, vol. 9, no. 5, pp. 1021-1035.

[9] K. Urahama and T. Nagao, 1995, "K-winners-take-all circuit with 0(N) complexity," IEEE Trans. Neural Networks, vol. 6, pp. 776-778.

[10] Z. Guo and J.Wang, 2011, "Information retrieval from large data sets via multiple-winners-take-all," Circuits and Systems (ISCAS2011) pp. 2669-2672.

[11] B. Sekerkiran, and U. Cilingiroglu, 1999, "A CMOS K-winners-take-all circuit with O(n) complexity," IEEE Trans. Circuits and Systems II, vol. 46, no. 1, pp. 1-5.

[12] Y. Hung, C. Y. Tsai, and B. D. Liu, "1-V rail-to-rail analog CMOS programmable winner-take-all chip with two-side searching capability for neurocomputing applications, 2003, " in Proc. Neural Networks and Signal Processing, vol.1, pp. 337-340.

[13] P.V. Tymoshchuk, 2013, "A fast analogue K-winners-take-all neural circuit," in Proc. Neural Networks (IJCNN), pp.1-8

[14] A. Kapralski, 1989, "The maximum and minimum selector SELRAM and its application for developing fast sorting machines," IEEE Trans. Computers, vol. 38, no. 11, pp. 1572-1577.

[15] M. Ogawa, K. Ito, and T. Shibata, 2002, "A general-purpose vector- quantization processor employing two-dimensional bit-propagating winner-take-all" IEEE Sym. VLSI Circuits Digest of Tech. Papers, vol. 35, no.11, pp. 244-247.

[16] A. Kapralski, 1989, "The maximum and minimum selector SELRAM and its application for developing fast sorting machines," IEEE Trans. Computers, vol. 38, no. 11, pp. 1572-1577

[17] M. Ogawa, K. Ito, and T. Shibata, 2002, "A general-purpose vector- quantization processor employing two-dimensional bit-propagating winner-take-all" IEEE Sym. VLSI Circuits Digest of Tech. Papers, vol. 35, no.11, pp. 244-247

[18] T. C. Hsu and S. D. Wang, 1997, "k-Winners-take-all neural net with O(1) time complexity". IEEE Trans. Neural Networks. vol. 8, no. 6, pp. 1557–1561.

[19] C. S. Lin, P. Ou, and B. D. Liu, 2001, "Design of k-WTA/Sorting Network Using Maskable WTA/MAX Circuit". In Proc. VLSI Symposium on Technology, Systems and Applications, pp. 69–72.

[20] H. Y. Li, C. M. Ou, Y.T. Hung, W. J. Hwang, and C. L. Hung, 2010, "Hardware Implementation of k-Winner-Take-All Neural Network with On-chip Learning," in Proc. Computational Science and Engineering, pp. 340-345.

[21] M. Yoon, 2015, "A Parallel Search Algorithm and Its Implementation for Digital k-Winners-Take-All Circuit", JSTS, vol. 15, no. 4, pp. 477-483.