

A VLSI High-Performance Encoder with Priority Lookahead

José G. Delgado-Frias and Jabulani Nyathi
*Department of Electrical Engineering
State University of New York
Binghamton, NY 13902-6000*

Abstract

In this paper we introduce a VLSI priority encoder that uses a novel priority lookahead scheme to reduce the delay for the worse case operation of the circuit, while maintaining a very low transistor count. The encoder's topmost input request has the highest priority; this priority descends linearly. Two design approaches for the priority encoder are presented, one without a priority lookahead scheme and one with a priority lookahead scheme. For an N -bit encoder, the circuit with the priority lookahead scheme requires only 1.094 times the number of transistors the circuit without the priority lookahead scheme. Having a 32-bit encoder as an example, the circuit with the priority lookahead scheme is 2.59 times faster than the circuit without the priority lookahead. The worst case operation delay is 4.4 ns for this lookahead encoder, using a 1- μ m scalable CMOS technology. The proposed lookahead scheme can be extended to larger encoders.

1 Introduction

Priority encoders are used in a number of computer systems and sub-systems. When several processes, modules, or units request a single hardware (or software) resource, a decision has to be made to allow a single request to use such a resource. The priority encoder implements a prioritized selection function where the resource is granted to the request with the highest priority according to the selection function. Examples of some sub-systems that use encoder functions include: bus [1], I/O [2], data comparator [3], and Interconnection Network Router [4]. A bus is used as a means of communication between modules. When more than one module request the bus, only one of them will get access to the bus. In this case, the top priority might be given to the processor. When an I/O device is requested by several modules, only one of these will be granted its request. Priority encoders in conjunction with comparators [3] can be used to compare two sets of parallel digital data having a large number of bits. An interconnection network router [4]

requires a priority selection function to ensure deterministic execution of routing algorithms and to select an output port that provides a short path.

Out of a number of requests a priority encoder selects only one of them to be served. This selection is based on a static or dynamic priority. A static approach has a fixed priority that can only be changed by explicitly changing the relative priority position of the requests. A dynamic approach allows changes in the priority at run time. These encoders can be implemented either in hardware for extremely fast processing or in software for flexible priority schemes. The proposed priority encoder accepts N input request lines and outputs only one active line corresponding to the request that has the highest priority. Priority encoding ensures that the input request with the lowest index has the highest priority. We explore a lookahead approach in order to minimize the propagation delay of the priority status during the worse case operation, hence increasing the operation speed of the circuit.

This paper has been organized as follows. In Section 2 we provide a description of two priority encoder approaches. Section 3 provides the VLSI circuit design of the two approaches, while Section 4 has simulation results of the proposed priority encoder. Some concluding remarks appear in Section 5.

2 Priority Encoder Scheme

In this section we introduce a novel priority encoder scheme. In this study the encoded Priority (EP) for a bit position i is expressed as:

$$EP_i = M_i \cdot P_i \quad (1)$$

where M_i is the priority encoder's input request and P_i is the priority status for this bit position. M_i and P_i are both one bit values. The operator (\cdot) in equation 1 and in the subsequent equations represents a logic AND, hence the encoded priority by definition is the result of ANDing the input with the priority status. The encoder implements a chained priority where inputs at lower bit positions have higher priority. The

priority status P_i is given by the equation:

$$P_i = \overline{M}_{i-1} \cdot P_{i-1} \quad (2)$$

where M_{i-1} and P_{i-1} are the request and the priority status, respectively, of the previous bit position. If there is no request at bit position $i-1$ (i.e $M_{i-1} = 0$), then the priority P_{i-1} is passed on to P_i .

Equation 2 shows that there is a concatenated dependence in order to determine the status of the priority bit P_i . Starting with P_0 we can write equations that describe how the priority bits are determined.

$$\begin{aligned} P_0 &= 1 \\ P_1 &= \overline{M}_0 \cdot P_0 = \overline{M}_0 \\ P_2 &= \overline{M}_1 \cdot \overline{M}_0 \\ P_3 &= \overline{M}_2 \cdot \overline{M}_1 \cdot \overline{M}_0 \\ &\vdots \\ P_{n-1} &= \overline{M}_{n-2} \cdot \overline{M}_{n-3} \cdot \overline{M}_{n-4} \cdots \overline{M}_1 \cdot \overline{M}_0 \end{aligned} \quad (3)$$

From the above set of equations P_0 has the highest priority, and the other priority bits have a dependence on the previous requests. If we allow \prod to represent a logical AND, we can write the above set of equations in a more concise form:

$$P_i = \prod_{j=0}^{i-1} \overline{M}_j \quad (4)$$

An N -input encoder requires $n-1$ ANDed terms to determine the priority status of the n^{th} entry. As N increases, a direct implementation of this approach will require a large number of gates as well as a large fan-out for request inputs such as M_0 . We therefore seek for ways to reduce the number of the ANDed terms in determining the priority status of any given entry. We explore the use of a 4-bit priority lookahead scheme, which shares some common characteristic with carry lookahead adders [5, 6].

We consider a case in which we have an N -bit priority encoder. We develop expressions for a priority lookahead (PL) status. The N -bit priority encoder is partitioned into 4-bit segments, hence the term 4-bit priority lookahead. The expressions that describe the 4-bit priority lookahead scheme are in equation 5.

$$\begin{aligned} PL_0 &= \overline{M}_3 \cdot \overline{M}_2 \cdot \overline{M}_1 \cdot \overline{M}_0 \\ PL_1 &= \overline{M}_7 \cdot \overline{M}_6 \cdot \overline{M}_5 \cdot \overline{M}_4 \\ &\vdots \\ PL_{m-1} &= \overline{M}_{4m-1} \cdot \overline{M}_{4m-2} \cdot \overline{M}_{4m-3} \cdot \overline{M}_{4m-4} \end{aligned} \quad (5)$$

where $m = \lfloor \frac{n}{4} \rfloor$. In the above set of equations it is clear that each 4-bit priority lookahead (PL) segment consists of four ANDed terms. The 4-bit priority lookahead scheme is used in conjunction with the set of equations in 3. In the event that $P_{i-1} = P_{i-2} = P_{i-3} = \cdots = P_0 = 1$ equation 1 becomes

$$EP_i = M_i \cdot \overline{M}_{i-1} \cdot \overline{M}_{i-2} \cdots \overline{M}_0 \quad (6)$$

Using the 4-bit priority lookahead scheme, we represent the encoded priority as:

$$EP_i = M_i \cdot \left(\prod_{j=1}^{i \bmod 4} \overline{M}_{i-j} \right) \cdot \left(\prod_{k=1}^{\lfloor \frac{i}{4} \rfloor} PL_{\lfloor \frac{i}{4} \rfloor - k} \right) \quad (7)$$

Equation 7 is an expression for all possible cases that can be encountered using the 4-bit priority lookahead scheme and based on this equation we can write the expressions for the encoded priority for all bit positions. The expressions provided in equation 8 below are for a 32-bit priority encoder.

$$\begin{aligned} EP_0 &= M_0 \\ EP_1 &= M_1 \cdot \overline{M}_0 \\ EP_2 &= M_2 \cdot \overline{M}_1 \cdot \overline{M}_0 \\ EP_3 &= M_3 \cdot \overline{M}_2 \cdot \overline{M}_1 \cdot \overline{M}_0 \\ EP_4 &= M_4 \cdot PL_0 \text{ since } PL_0 = \overline{M}_3 \cdot \overline{M}_2 \cdot \overline{M}_1 \cdot \overline{M}_0 \\ &\vdots \\ EP_{31} &= M_{31} \cdot \overline{M}_{30} \cdot \overline{M}_{29} \cdot \overline{M}_{28} \cdot PL_6 \cdot PL_5 \cdot \\ &\quad PL_4 \cdot PL_3 \cdot PL_2 \cdot PL_1 \cdot PL_0 \end{aligned} \quad (8)$$

Use of the 4-bit priority lookahead scheme can be extended to a second level by grouping the 4-bit priority lookahead into two sets, to form a 16-bit priority lookahead scheme. The expression for the second level priority lookahead scheme appears in equation 9.

$$\begin{aligned} PL_0^2 &= PL_3 \cdot PL_2 \cdot PL_1 \cdot PL_0 \\ PL_1^2 &= PL_7 \cdot PL_6 \cdot PL_5 \cdot PL_4 \end{aligned} \quad (9)$$

The encoded priority at bit position 31, when using the second level priority scheme becomes:

$EP_{31} = M_{31} \cdot \overline{M}_{30} \cdot \overline{M}_{29} \cdot \overline{M}_{28} \cdot PL_6 \cdot PL_5 \cdot PL_4 \cdot PL_0^2$
Having described the priority encoder scheme, we now present the VLSI circuits of the proposed design. In the next section an implementation of the equations provided in this section will be presented.

3 Priority Encoder VLSI Circuits

In order to show how the proposed priority encoding approach can be implemented, we have designed

and simulated a CMOS priority encoder. In this implementation, we have used pass transistor logic and dynamic circuitry not only to achieve high performance but also to minimize silicon real estate. In this section, we describe the circuitry of a CMOS priority encoder with no lookahead. Then extremely fast priority circuitry is added to this circuit, with a very small impact on the overall number of transistors, to produce an encoder that uses a priority lookahead scheme.

3.1 Basic Priority Encoder Cell

Figure 1 shows the unit cell of the priority encoder described by the expression $EP_i = M_i \cdot P_i$. This circuit requires a precharge phase before the request signal (M_i) is processed. During this phase the $\overline{M'_i}$ and P_i signals are precharged to logic 1 by transistors T_{pchr} and T_{pchp} respectively. The transistor T_{pchp} is an n-type transistor, therefore, the priority status can only be precharged to $V_{dd} - V_{tn}$, instead of V_{dd} (where V_{tn} is the threshold voltage of an n-type transistor). This reduces the time required to discharge P_{i+1} when M_i is set to logic 1. The signal labeled as M'_i sets the status of transistor T_{pdp} . This transistor is used to pull down the priority; thus, at precharge transistor T_{pdp} is set off to prevent any discharge of the priority signal P_{i+1} . The priority status is normally precharged to logic 1, in order to avoid delays in propagating $P_i = 1$. Once the precharging of $\overline{M'_i}$ and P_i has been completed, the request (M_i) is passed when the clock signal is set to logic 1. Transistor T_{lch} allows the request to be dynamically latched on the gates of the inverter and transistor T_{pp} ; thus, the duration of the clock pulse could be made extremely short. If the current request has the highest priority and $M_i = 1$, it implies that the priority status to be propagated is at logic 0, therefore, P_{i+1} which is normally precharged, must be discharged. The circuit arrangement of Figure 1 indicates that this is performed by using the value of M'_i , to turn on the pull down transistor (T_{pdp}) and hence discharge P_{i+1} . When $M_i = 1$ transistor T_{pp} is turned off by the signal $\overline{M'_i}$ to prevent a discharge of priority P_i . Figure 1 shows how the expression $EP_i = M_i \cdot P_i$ is implemented, if the current request is active and the current priority status has not been discharged, then the corresponding encoded priority (EP_i) gets set to logic level 1.

Figure 2 shows how a number of priority encoder cells can be put together to implement a large encoder. This figure shows a four-input encoder unit in which four basic cells are cascaded; the priority status is conditionally propagated to the lower cells. If M_0 is at logic 0, then P_0 gets propagated to the next cell, and to minimize the delay of propagating a “1”,

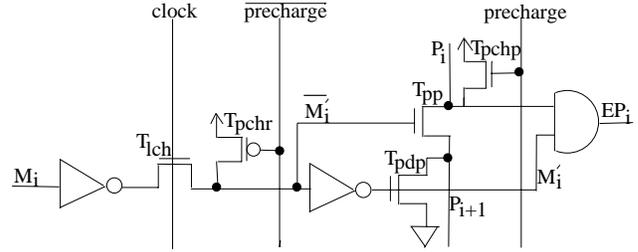


Figure 1: Priority encoder unit cell.

we precharge P_1 . If request M_0 is at logic level 1, P_0 does not get propagated, instead P_1 gets discharged to logic 0. This circuit operation satisfies the expression; $P_i = \overline{M_{i-1}} \cdot P_i$, in Section 2, and shows the dependence of the current priority status on the logic value of the previous priority status. P_0 is connected to the power supply to conform with the first expression of equation 3 (i.e $P_0 = 1$), giving the priority status an initial value of logic 1.

The priority encoder unit shown in Figure 2 can be extended to accommodate a larger number of inputs. We have included a buffer per four-input encoder unit to reduce the delays in the priority chain. Assuming a 32-bit decoder, that uses eight, four-input encoder units, the critical operation occurs when $M_0 = 1$ and $M_{31} = 1$ with requests M_1 through M_{30} all at logic 0. This results in $EP_0 = 1$, and since M_0 has the highest priority, EP_{31} must be set to logic 0. This is achieved by setting P_1 to logic 0 and propagating this status to the last entry. This represents the worst case operation because P_1 must be propagated through a series of n-type transistors, in order to set P_2 through P_{31} to logic 0. Equation 3 in Section 2 shows a recursive relation between P_1 and P_{31} , and this is represented in Figure 2 by a chain of pass transistors, whose gates are driven by $\overline{M'_i}$ (transistors represented by transistor T_{pp} in the unit cell) and a pull down transistor whose gate is driven by the requests M'_i (transistors that correspond to transistor T_{pdp} in the unit cell). The priority status is required to determine the logic value of the encoded priority, as a result it must be propagated with minimal delay to the lower entries, and for the worst case operation described, the propagated priority status is required to set EP_{31} low, since a request with the highest priority has been encountered. Due to the way the priority propagates in this scheme, the delay for the worst case operation tends to be long. This in turn may impose a severe restriction on the clock of the system, where this encoder is used, since the clock should allow enough time to process all the cases (including the worst case).

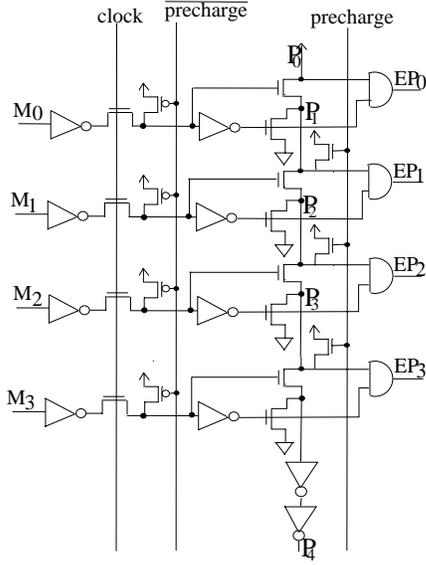


Figure 2: 4-bit priority encoder.

3.2 Priority Encoder Cell with Lookahead

In order to reduce priority status propagation delays, in this section we propose a circuit that implements the priority lookahead in a simple, but yet effective manner. Figure 3, shows the basic cell for the priority lookahead scheme. In this figure the unit cell from Figure 1, has been extended to include a lookahead line and a pull-down transistor (T_{pdl}) whose gate is driven by the request. At precharge time, the lookahead line is set to logic 1. When the clock goes to logic 1 and if $M_i = 1$, transistors T_{pdl} and T_{pdp} conduct discharging PL_j and P_{i+1} respectively. Cascading several basic cells of Figure 3, results in a chain of pass transistors (T_{pp}), and a delay increase in propagating $P_i = 0$. The priority lookahead scheme provides a fast path for P_i to propagate to other cells through the lookahead line (PL_j). The signal on the priority lookahead line propagates to the lower cells much faster than the priority signal that propagates through the pass transistors (T_{pp}).

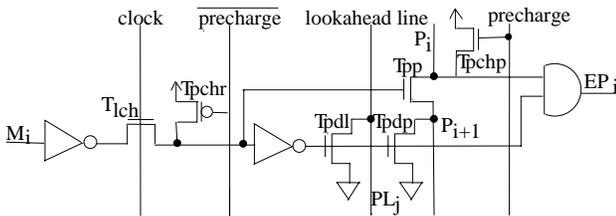


Figure 3: Priority encoder with lookahead (basic cell).

Figure 4 shows a 4-bit priority lookahead encoder. We use T_{pch} , an n-type transistor, to precharge the priority lookahead line. This reduces the time required to pull-down PL_j if a single input request within the four-input unit is at logic 1. To determine the status of the priority lookahead line, all the requests M_0 through M_3 are considered. The logic value of the priority lookahead line in this case is: $PL_0 = \overline{M_0} \cdot \overline{M_1} \cdot \overline{M_2} \cdot \overline{M_3}$, and the general expression for an N -bit priority encoder appears in equation 5, Section 2. If request M_0 has the highest priority, and the rest of the requests (M_1 through M_3) are at logic 0, then, the priority lookahead line gets discharged by transistor T_{pdl} corresponding to the first entry and a logic 0 gets propagated to the next four entries. The signal on the priority lookahead line gets inverted, turning transistor T_{pdpc} on, resulting in a logic 0 being propagated upwards through the chain of series pass transistors, on the other hand the priority status, $P_i = 0$, is being passed through the same chain of pass transistors, downwards. This approach significantly reduces the delay of propagating a logic 0 within the four-input encoder unit.

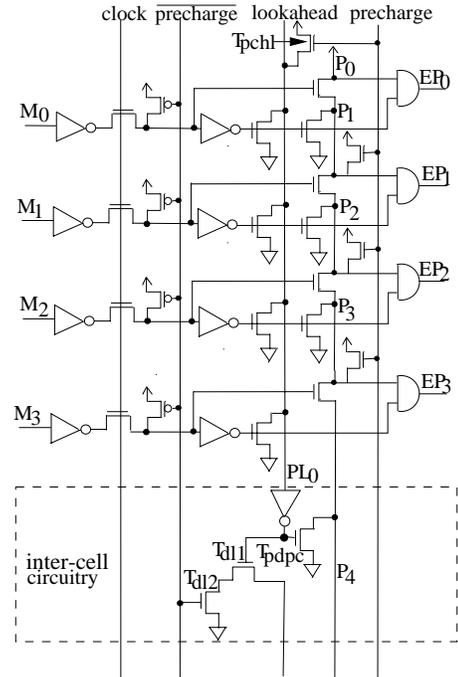


Figure 4: 4-bit priority lookahead encoder.

The 4-bit priority encoder can be cascaded to produce an N -input encoder. Cascading the 4-bit encoder unit requires that some additional circuitry be included between the modules, and this inter-module

circuitry appears in the dotted box in Figure 4. The inter-module circuitry consists of three n-type transistors two of which are in series, with transistor T_{dl1} driven by the inverse of the signal on the priority lookahead line and transistor T_{dl2} being driven by the *precharge* signal. Transistor T_{dl1} could have been used as the pull-down transistor, however, transistor T_{dl2} is required to accommodate the inverter delays. Due to the delay that occurs when several modules are put together, the precharge cycle of PL_{j+1} could begin while transistor T_{dl1} is still conducting. This in turn may not allow the PL_{j+1} priority lookahead line to be charged properly. The transistor labeled as T_{pdpc} within the inter-module circuitry serves to discharge the priority status of the next module as well as that of the previous module, if a request with the highest priority has been encountered in any of the entries above. For all the entries that give *i mode 4* = 0, transistor T_{pdpc} serves to discharge the priority status P_{i+1} , and this results in a logic 0 being propagated bidirectionally to P_i and P_{i+2} , assuming that input requests M_i and M_{i+1} are at logic 0. The 4-bit priority lookahead scheme depicted in Figure 4, can be extended to a second level priority lookahead scheme, by grouping the 4-bit cells into groups of sixteen and adding a priority lookahead line. The status of a single second level priority line PL_j^2 is determined by the status of any or all of the four first level priority lookahead lines. After the precharge cycle has been completed, any of the PL_j 's that is at logic 0, will cause PL_j^2 to be discharged.

4 Simulation Results

The circuits described in the previous section have been simulated for functionality and performance using SPICE. A 32-bit encoder has been designed and tested using a 1- μ m scalable CMOS technology. Before reporting the results of the simulations it is necessary to point out that we have arranged the transistor geometries to achieve better performance. To minimize degradation of the gain factor (β), for the series transistors (T_{pp} in Figure 3) we use a gate ratio (width:length) of 4:1. When a high priority request is found the inverter on the priority lookahead line must switch its output very fast from a logic 0 to a logic 1. The p-type transistor within this inverter must accomplish this switching; thus, we have to ensure that this transistor's gain factor is high enough to provide the desired switching speed.

When an encoded priority (EP_i) in the upper entries has been set to logic 1, priority status P_{i+1} gets set to logic 0, and propagated to the lower entries, to ensure that the rest of the encoded priorities are set

to logic 0, since a request input (M_i) with the highest priority has been encoded. For the critical operation, priority status P_{n-1} , must be set to logic 0 in order to prevent EP_{n-1} from being set to logic 1. In this case $P_1 = 0$ needs to be propagated to the last entry. This critical (or worst) case is simulated and the results are reported below. We measure the propagation delay for the worst case operation, as the time it takes for priority status P_{n-1} to reach 10% of V_{dd} , once the clock reaches 90% of V_{dd} . The SPICE simulation results show that the circuit with the priority lookahead scheme performs much better than the circuit without the priority lookahead scheme. Figure 5 depicts the SPICE simulation results of the circuit without the priority lookahead, while in Figure 6 we display the results of the circuit that uses the lookahead scheme. It takes 11.4 ns to propagate the priority status if the first design approach is considered, and 4.4 ns when the priority lookahead approach is used. The percent improvement can be computed by first considering the ratio of the delays:

$$\text{Delay Ratio} = \frac{\text{Encoder w/o PL}}{\text{Encoder w/ PL}} = \frac{11.4 \text{ ns}}{4.4 \text{ ns}} = 2.59$$

This shows that the design with the priority lookahead scheme is 159% faster than the design without the priority lookahead.

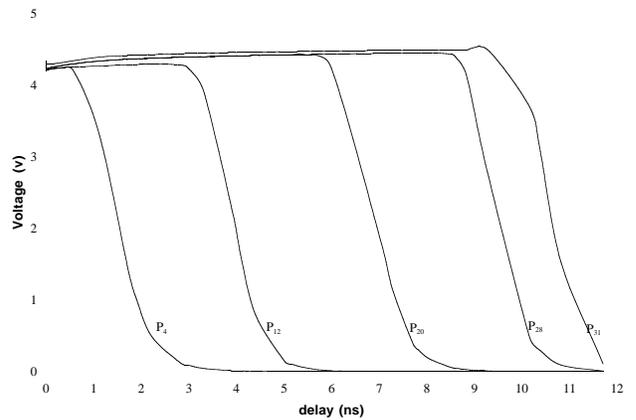


Figure 5: Propagation delays of a 32-bit encoder with no priority lookahead scheme.

To obtain better performance we have added circuitry to perform the lookahead function. The transistor count for an N -bit encoder using the encoder without the priority lookahead is given by: $15N + 4 \cdot \lfloor \frac{N}{4} \rfloor$ and for that with the priority scheme the transistor count is: $16N + 6 \cdot \lfloor \frac{N}{4} \rfloor$. Based on these expressions, we compute the percent increase in transistor count by considering the ratio of the number of transistors given an N -bit encoder, as follows:

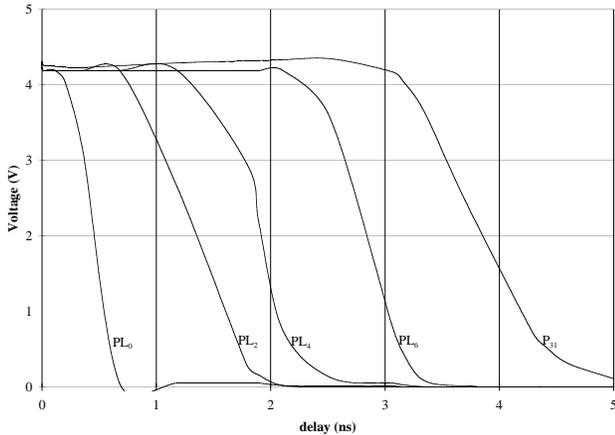


Figure 6: Propagation delays of a 32-bit encoder using a 4-bit priority lookahead scheme.

$$\text{Transistor Ratio} = \frac{\text{Encoder w/ PL}}{\text{Encoder w/o PL}} = 1.094$$

Therefore the increase in transistor count is only 9.4%. This is a small increase compared to the delay reduction. If we consider a 32-bit encoder, we have that the circuit without the priority lookahead scheme takes longer (11.4ns), to propagate the priority status during the worst case operation, while it takes 4.4 ns to propagate the priority status using the circuitry, that has a priority lookahead scheme.

5 Concluding Remarks

In this paper we have proposed a novel priority encoder scheme and its VLSI circuitry. Our priority encoder uses a fixed priority scheme to encode N -input requests. The encoder receives N -inputs which could either be at logic 0 or 1, encodes them, and sets a single output to logic 1. The fixed priority scheme ensures that the output corresponding to the input request that has the highest priority gets set to logic 1 while the rest of the outputs are set to logic 0, irrespective of the logic level of their corresponding input requests. For a linear N -input encoder, the input with the least index (M_0) if set to logic 1, has the highest priority. We have shown two design approaches one of them uses a novel priority lookahead scheme. The circuit for the priority encoder that uses a priority lookahead scheme, requires an additional transistor and a lookahead line as additions to the basic cell of the encoder without the priority lookahead. The priority lookahead line enables us to group the basic cells in groups of four, forming a 4-bit priority lookahead circuit. This permits the priority status (P_i) to be propagated to the cells below much faster. This scheme can be applied recursively for larger encoders. The priority

lookahead cell circuit has only 1.094 times, more transistors than the circuit without the priority lookahead scheme. A 32-bit priority encoder and a 1- μm CMOS technology have been used as means to show the potential of the proposed encoder scheme. SPICE simulations have shown that the design that uses the priority lookahead scheme outperforms the design without the priority lookahead scheme by 159%, when the worst case operation of both circuits is considered. A 32-bit encoder without the priority lookahead has a delay of 11.4 ns while one with the priority lookahead has a 4.4 ns delay. In a sub-micron technology, such as 0.25- μm , the proposed priority encoder would be greatly improved.

Our priority lookahead encoder is an extremely fast general purpose encoder that can find applications in a number of digital computer systems and sub-systems. The priority lookahead scheme is an important feature of this design, since it significantly reduces the priority status' propagation delay. The proposed scheme can be applied recursively for larger encoders. The 4-bit priority encoder circuitry can easily be extended to a second level priority lookahead scheme, while still maintaining its high performance.

References

- [1] E. D. Adamides *et al*, "Cellular Logic Bus Arbitration," *IEE Proceedings. Part E, Computers and Digital Techniques*, Vol. 140, No. 6, pp. 289-296, November 1993.
- [2] P. H. Garrett, *Advanced Instrumentation and Computer I/O Design -Real-Time System Computer Interface Engineering*, IEEE Press, 1994.
- [3] S. Murugesan, "Use Priority Encoders for Fast Data Comparison," *Electronic Engineering*, Vol. 42, p. 24, July 1989.
- [4] D. H. Summerville, J. G. Delgado-Frias, and S. Vassiliadis, "A Flexible Bit-Pattern Associative Router for Interconnection Networks," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 7, No. 5, pp. 477-485, May 1996.
- [5] J. B. Kuo *et al*, "A BiCMOS Dynamic Carry Lookahead Adder Circuit for VLSI Implementation of High-Speed Arithmetic Unit," *IEEE Journal of Solid - State Circuits*, Vol. 28, No. 3, pp. 375-378, March 1993.
- [6] T. Lynch and E. E. Swartzlander, "A Spanning Tree Carry Lookahead Adder," *IEEE Transactions on Computers*, Vol. 41, No. 8, pp. 931-939, August 1992.