

A WAVE-PIPELINED CMOS ASSOCIATIVE ROUTER FOR COMMUNICATION SWITCHES

José G. Delgado-Frias and Jabulani Nyathi

Department of Electrical Engineering
State University of New York
Binghamton, NY 13902-6000

ABSTRACT

A wave-pipelining approach is used to improve the performance of a VLSI router. Wave-pipelining has a potential of significantly reducing clock cycle time and silicon real estate. The design approach considered in this paper allows data propagation between stages to occur without the use of intermediate latches. Control signals are designed to ensure that intermixing of *data waves* does not occur. This study's results show that using wave-pipelining reduces the clock period. The circuit delays become the limiting factor, preventing further clock cycle time reduction.

1. INTRODUCTION

A critical component in computing and communication systems is the router which receives, forwards, and delivers messages. The router system transfers messages based on a routing algorithm which is a crucial element of any communication network. Given the reconfiguring requirements for many computing systems, a number of routing algorithms as well as network topologies must be supported, hence the need for a very high performance flexible router to support these requirements. To maximize a machine's overall performance and to accommodate reconfiguration in a distributed environment requires matching the application characteristics with a suitable routing algorithm and topology. It is of extreme importance that the routing algorithm execution time be extremely short. This time dictates how fast a message can advance through the network, since a message cannot be transferred until an output port is selected by the routing algorithm. Thus, the routing algorithm execution time must be reduced to decrease message delays.

The use of latches in pipelines limits the reduction of the clock period due to the additional time during which the latch is sensitive to any change in the data input. In any pipeline system, the clock period is set by longest delay produced by the combinatorial logic and wiring paths between input and output latches of a given data path pipeline stage [1]. These latches need be synchronized using a clock;

as clock frequencies increase clock distribution becomes a more difficult task. Wave-pipelining offers the potential for high performance, since its maximum clock frequencies are limited by the path delay differences as opposed to the longest path delays [1, 2]. This approach provides a method for significantly reducing clock cycle time, silicon real estate, power, and latency. The waves of electrical signals are used to realize a pipeline [1], and to isolate one pipeline stage from the adjacent ones the delays due to the logic and wiring paths are used. Wave-pipelining could potentially increase the pipeline rate without using additional latches.

In this paper, a novel VLSI router system is presented; this system provides flexibility to execute a number of routing algorithms as well as extremely fast execution times. This paper is organized as follows: Section 2 describes the router architecture and its major circuits. Section 3 presents the wave-pipelining approach as well as some signals of importance. Some concluding remarks appear in Section 4.

2. BIT-PATTERN ASSOCIATIVE SCHEME

The bit-pattern associative router scheme provides hardware support to run intrinsic routing algorithms. In routing algorithms the network topological characteristics are usually reflected in the network addressing scheme.

2.1. VLSI Organization

A novel dynamic content addressable memory (DCAM) cell has been designed as part of the bit-pattern router VLSI implementation [3]. The DCAM design includes per entry unique bit masking to provide the parallel evaluation of all the bit-patterns on the input data. The design uses dynamic storage cells to hold the matching data condition. This condition requires a ternary digit (with values: 0, 1, and X-don't care) per comparison bit. The DCAM cell is significantly smaller than the static cell while maintaining the same level of performance and functionality.

The organization of the pipelined router is shown in Figure 1. The CAM cells perform the bit-comparison between

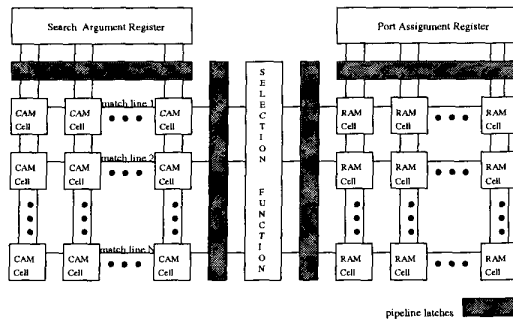


Figure 1: Bit-pattern router organization.

the stored patterns and the argument (destination address, network status and algorithm parameters) sent by the search argument register (SAR). Once a comparison with all the stored patterns has been performed the condition of the match lines is latched; this accommodates the requirements for a pipeline. If more than one match line has been selected, the selection function circuitry allows only one selected line to pass to the port assignment memory (RAM cells). Then, the selected row of RAM cells is read and passed to the port assignment register. In order to provide programmability to the dynamic CAM and RAM cells, row select shift registers (not shown in Figure 1) are used. The row select shift register generates signals (row 1...row n) to select a row in the CAM array to write to.

2.2. Dynamic CAM Cell

The DCAM cell implements a comparison between an input and the ternary digit condition stored in the cell. The proposed cell has a small transistor count. The circuit of a single DCAM cell, shown in Figure 2, consists of eight and a half transistors; transistor T_e is shared by two cells. The read, write, evaluation and match line signals are shared by the cells in a word while the BIT_STORE, NBIT_STORE, BIT_COMPARE and NBIT_COMPARE lines are shared by the corresponding bit in all words of the matching unit. The design uses a precharged match line to allow fast and simple evaluation of the match condition. This condition is represented by the DCAM cell state which is set by $Sb1$ and $Sb0$ node status. This state is stored in the gate capacitance of the transistors T_{c1} and T_{c0} . In Table 1 the possible DCAM cell stored values are listed. Transistors T_{ref0} and T_{ref1} are used for refreshing.

The DCAM cell performs a match between its stored ternary value and a bit input (provided by BIT_COMPARE and NBIT_COMPARE which represent a bit and its opposite value, respectively). The match line is precharged to "1"; thus, when a no match exists this line is discharged by means of transistor T_m . Match operation mode involves

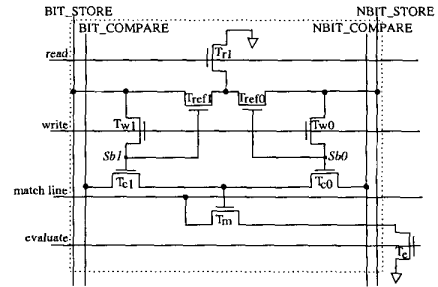


Figure 2: CMOS circuit for a DCAM cell.

Table 1: Representation of stored data in a DCAM cell.

$Sb1$	$Sb0$	state
0	0	X(don't care)
0	1	1(one)
1	0	0(zero)
1	1	not allowed

comparing the input data to the patterns stored in the DCAM and determining if a match has been found. During match operation, the input data is presented on the BIT_COMPARE line and its inverse value on the NBIT_COMPARE line. Before the actual matching of these two values is performed, the match line is precharged to "1" which indicates a match condition. The matching of the input data and the stored data is performed by means of an exclusive-or operation which is implemented by the two transistors (T_{c1} and T_{c0}) that hold the stored value.

2.3. Selection Function and RAM

The selection function should be designed to ensure the deterministic execution of the routing algorithms. With a given input (i.e. destination address) and a set of patterns stored in the matching unit, the port assignment should always be the same. The priority allows only the highest priority pattern that matches the current input to pass on to the port assignment memory. The encoded priority (EP_i) output depends on the match at the current bit-pattern and the priority for this row. If both match and priority are "1"; then the encoded priority is true. The priority lookahead scheme has been proposed and implemented; it has been reported in [4].

The port assignment memory or RAM holds information about the output port that has to be assigned after a bit-pattern that matches the current input is found. This memory is proposed to be implemented using a dynamic approach. The selected row address is passed from the priority encoder. The cells in this row are read and their data is latched in the port assignment register. The DRAM structure is able to perform an OR function per column when multiple RAM rows are selected; this is when multiple matches

are passed on. The DRAM structure is explained in [3].

3. WAVE-PIPELINED VLSI ORGANIZATION

In order to increase the performance of the proposed associative router, a wave-pipelined architecture is studied and simulated. Wave-pipelining has the potential of significantly reducing the clock cycle time, the area associated with inter-stage latches, power, and latency [1, 2]. As clock rates continue increasing, clock distribution is becoming a major problem; wave-pipelining could help to alleviate it by significantly reducing the clock load as well as the number of inter-stage latches [1].

3.1. Wave-Pipelined Router Requirements

Wave-pipelining involves propagation and manipulation of data from the input register to the output register without any intermediate latches. Wave-pipelining allows for several unrelated *data waves* to be within the system each at a different stage. Removal of intermediate latches introduces the possibility that two or more *data waves* can be intermixed. A wave-pipelined design must then ensure that data intermixing does not occur. We have designed circuitry that ensures that data propagates within its own wave. The signals from these circuits allow data to ripple to the next stage without any intermixing. The first signal of importance is the signal used to precharge the match line. The requirement for precharging the match line is that this occurs before evaluation takes place and after the output of the match line has been passed to the next stage. We, therefore, design the precharge signal using the signals used to evaluate and pass the match line output to the next stage. At evaluation time it is required that the match line precharge signal be at "1", to prevent a conflict of operations on the match line. Also care must be taken to prevent precharging the match line before its output is passed to the next stage. The match line precharge signal, therefore, has to be low only when both the evaluate and pass signals are at "0".

The evaluate signal must go to "1" after the data passed to the DCAM for comparison has stabilized on the bit lines (BIT_COMPARE and NBIT_COMPARE). There is circuitry that generates this signal once all the lines have been set to their appropriate input values. This circuitry shown in Figure 3 takes into account setting a "0" or "1" in the particular CAM array. Its operation is governed by the status of the match line. The inverse of the clock gets passed to an intermediate node when the the match line is at "1" and this intermediate node gets precharged to "1" whenever the match line is at "0". The inverse of the intermediate node is the evaluate signal. Using the match line to pass the clock signal and set the evaluate signal to "0" when evaluation is completed provides the proper duration for the evaluate

signal to stay at "0" or "1". Once evaluation has been completed the match line outputs are passed to the priority encoder. Thus, the pass signal must immediately be active following completion of the evaluation process. The circuit used to generate the pass signal is shown in Figure 4. The pass signal is designed to mimic the path that a zero on the match line (non-matching condition) takes once evaluation completes. Passing a "0" to the priority encoder provides the maximum delay that can be experienced in passing the matching unit's outputs to the priority encoder and, therefore, constitutes the worst case propagation delay for this operation. The duration and voltage level of both the evaluate and pass signals need be just long and high enough to accomplish their purpose. The circuits in Figures 3 and 4 have been designed to sense the duration and voltage levels of these signals.

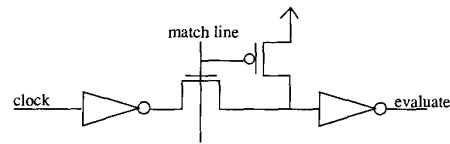


Figure 3: Generating the evaluate signal.

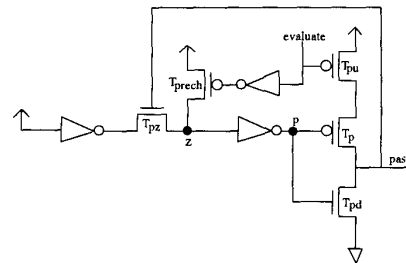


Figure 4: Generating the pass signal.

3.2. Wave-Pipelining Signals

The signals generated by the above circuits appear in Figure 5 along with the clock. Once the output of the match lines have been received by the hit and priority logic (priority encoder) a decision needs to be made when more than one matching condition has been received. The priority encoder is designed to propagate a priority status (P_i) to the entry below it indicating whether it has registered a match. Some of the signals of importance in this scheme are shown in Figure 6. The selection function's critical operation occurs when the first and last entries of the priority encoder simultaneously receive inputs indicating that matching conditions have been found in the corresponding matching unit

entries. The first entry of the priority encoder has to propagate a "0" to the last entry of the priority encoder, to prevent generation of a pointer to the DRAM by the last entry. In Figure 6 we show signals used to enable the DRAM pointers for the first and last entries from a setup in which the last entry always finds a match and the first entry finds a match every other clock cycle. We show the delays involved. The plots in Figure 7 are those of the DRAM pointers to the first and last entries of the DRAM. They serve to show that an output port assignment can be read from the DRAM array every one and a half clock cycles.

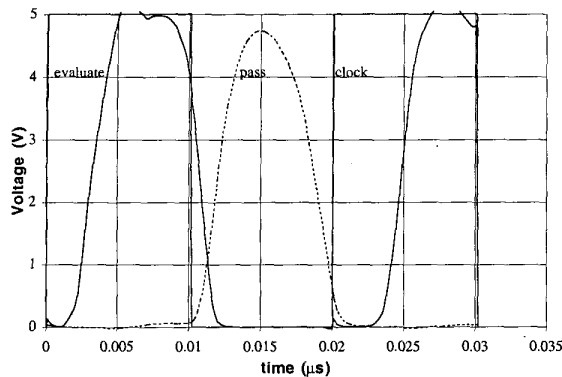


Figure 5: Plot of the Matching Unit control signals.

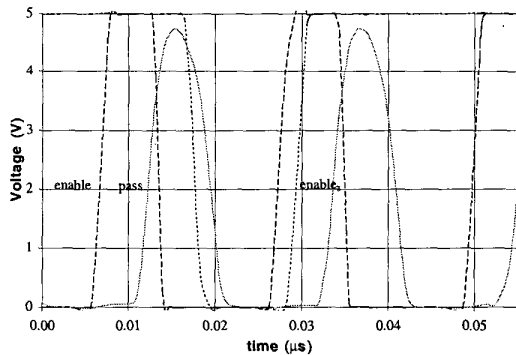


Figure 6: Plots of the enable signals.

4. CONCLUDING REMARKS

An extremely fast flexible router is very useful since it is able to accommodate a number of routing algorithms and networks. The execution of the routing algorithm is critical in all the network communications because messages or packets cannot be sent until the destination is determined. To realize smart interconnects it is necessary to execute dif-

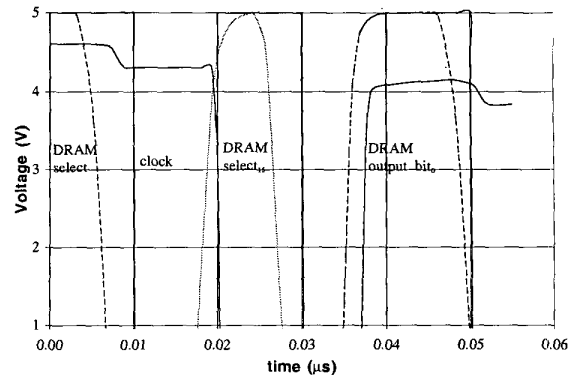


Figure 7: Plot of the DRAM pointers.

ferent algorithms that change due to load of the network or the application being run. Thus, there is a need for an extremely high performance reconfigurable router such as the one described in this study.

In this paper we have presented a novel wave-pipelined router that is capable of executing routing algorithms in one and a half cycles. As mentioned earlier, wave-pipelining has the potential for increasing further the performance of computer systems [1]. Issues such as signal generation, uneven delays, and timing have been addressed to synchronize the pipeline. This is one of the first studies that attempts to address these issues using an implementation of a somewhat complex pipelined architecture as an exploratory means to gain knowledge in this field. The system has been implemented using a CMOS technology.

5. REFERENCES

- [1] W. P. Burlinson, M. Ciesielski, F. Klass, and W. Liu, "Wave-Pipelining: A Tutorial and Research Survey," *IEEE Trans. on VLSI Systems*, vol. 6, no. 3, pp. 464-474, Sept. 1998.
- [2] C. T. Gray, W. Liu, R. K. Cavin, III, "Timing Constraints for Wave-Pipelined Systems," *IEEE Trans. on Computer-Aided Design*, vol. 13, no. 8, pp. 987-1004, Aug. 1994.
- [3] J. G. Delgado-Frias, J. Nyathi and D. H. Summerville, "A Programmable Dynamic Interconnection Network Router with Hidden Refresh," *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol. 45, no.11, pp. 1182-1190 Nov. 1998.
- [4] J. G. Delgado-Frias and J. Nyathi, "A High Performance Encoder with Priority Lookahead," *IEEE Sixth Great Lakes Symposium on VLSI*, pp. 59-64, Lafayette, Louisiana, Feb. 1998.