Self-Timed Pipeline with Adder

John Compton and Alexander Albicki

Dept. of Electrical Engineering
University of Rochester
Rochester, NY 14627

Abstract

This paper documents the design of an asynchronous pipeline structure comprising a ripple carry adder and registers placed before and after the adder. A scheme was created for monitoring the ripple of the carry through the adder. This scheme provides a means of determining when the addition is complete. The design approach uses transmission gate logic throughout. Results of SPICE simulation on the various building blocks of the circuit are presented.

Motivations

David, Ginosar, and Yoeli [1] discuss self timed combinational circuits using dual rail logic. In this approach, each bit is represented by two wires: the logical states of the the two wires encode to 0, 1, and undefined. A function implemented in dual rail logic requires both the normal function and a complementary function, effectively doubling the size of the circuitry. On top of this, there is additional circuitry required to detect completion of the operation.

Sutherland's [2] approach to asynchronous logic utilizes handshake signals between elements of the system. Rather than level sensitive signals for handshaking, the use of transition signalling is proposed: e.g., if a transition occurs on an acknowledge handshake line, then the acknowledge is said to have occurred. Muller C elements provide control for the latches in Sutherland's scheme.

Due to the conceptual simplicity of Sutherland's approach we decided to use it as an overall framework.

One thing that Sutherland does not go into in detail is how to derive self-timing information from combinational circuits. The diagrams in his paper merely show delay elements associated with the combinational logic blocks. It is conceivable that the dual rail approach and transition signalling could be combined to provide self timed combinational circuits along with efficient communication of signals but we decided to pursue a different route, discussed in the next section.

Completion detector scheme

The combinational logic block in this project is a ripple carry adder. Because of the way the carry ripples through the adder blocks, the amount of time it takes to add two numbers depends on the number of carries required. The worst case is obviously the case where carry ripples through every block of the adder. Note that the carry may ripple by going high, or, if the previous addition generated carries, the carry may ripple by going low. Hence, both transitions of the carry mean the subsequent stage must change its outputs. The carry chain of the ripple carry adder is the critical timing path of the adder, and the one which should be monitored.

We designed a transition detector which provided a pulse any time a carry output changed state and remained in its new state (this avoids problems with static logic hazards). Essentially the detector is an exclusive-OR gate with both inputs driven by the carry output; however, one of the inputs is delayed by means of a slow inverter (Figure 1).

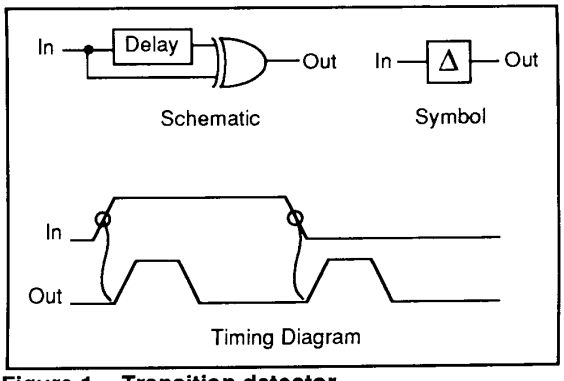

Figure 1 - Transition detector

0-8186-2610-0/92 $3.00 © 1992 IEEE

When a transition detector is triggered it operates a latch, holding the previous value of the request signal. When the transition detector times out, the latch passes the request signal through. When the request signal makes it all the way through the latches, then the addition is complete and the output register may store the result. The path of the request signal through the latches and the control of the latches by transition detectors is shown in Figure 2 for a 3 bit adder. The delay block at the bottom of the figure allows time for the carry out of the first adder block to occur and the transition detector to go high as a result before the request signal reaches the first latch. Figure 3 is a timing diagram showing how the request transition propagates through the latches.

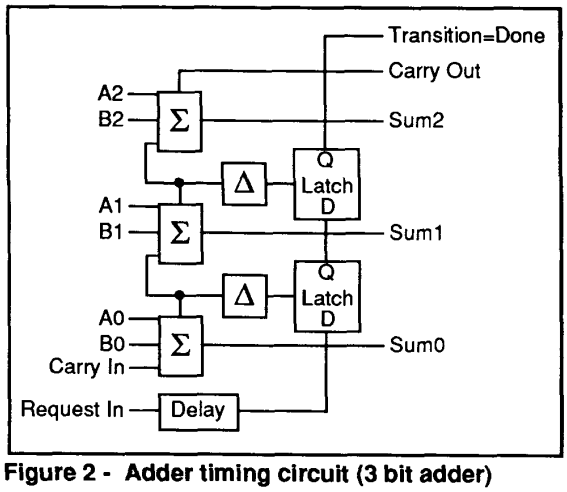

Figure 2 - Adder timing circuit (3 bit adder)

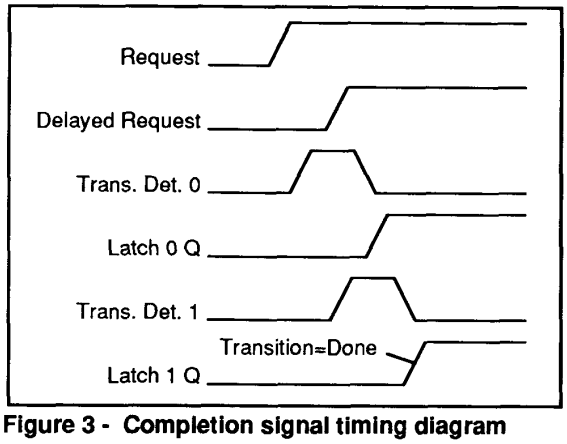

Figure 3 - Completion signal timing diagram

Figure 4 shows a 4 bit adder with input and output pipeline registers. This figure shows how the transition from the last latch in the adder timing circuit causes the output pipeline register to hold the result of the addition and also allows the input pipeline register to accept the next input value.

Circuit design

There are six building blocks used in this project: the transition detector, the one bit adder, the latch for the control of the completion signal, a fixed delay element, the capture pass latch used in the pipeline registers, and the Muller C element.

The transition detector has several challenging design goals: the detector should respond quickly since it is part of the ripple path; it should be easy to change the pulse width to match the response time of the carry portion of the adder; and the timing characteristics of the detector should be the same whether a rising edge or a falling edge is detected.

Figure 5 shows our implementation of the transition detector. Inverters I4 and I5 along with transmission gates T2 and T3 form an exclusive-NOR gate; inverter I6 turns this into an exclusive-OR gate and buffers the output. I1 buffers the input and provides drive for the transmission path through T1 and T2 to I6 and for the inverters I2 and I4. I2 is a slow inverter and provides the delay indicated in the schematic in Figure 1. Transmission gate T1 is always enabled and matches the delay which inverter I4 introduces into the lower transmission path; T4 provides a similar function for matching the delay of I5. The transfer characteristics of inverters and transmission gates are not identical, however. They behave similarly for a quickly changing signal, but for a slowly changing signal the transmission gate will pass the signal faithfully whereas the inverter will reach a threshold and then change somewhat abruptly. Therefore it was necessary to add inverter I3 to buffer the slowly changing output of the delay inverter I2; this provides a sharper transition at the inputs to I5 and T4. As indicated in the chart in Figure 5, the sizes of devices in the circuit take into account the mobility difference between PMOS and NMOS materials. By taking into account the differences in mobility and by equalizing the path delays in the transmission and control paths, the circuit was made to respond in a similar manner for both rising and falling transitions of the input.

Note that there is no need for a transition detector after the last addition block; when the transition detector for the next to last block times out, the last block has completed its job. Hence, there would be an odd number of transition detectors and latches for an even number of bits. In order to minimize propagation delay through the latches, we used an inverting latch thereby avoiding an additional inverter in each latch. If there are an even number of bits

in the adder, an inverter needs to be put at the end of the odd numbered chain of latches.

Our initial design of the circuit elements used transmission gate logic and fully static design. This conservative design approach provided a certain structural regularity in the layout of the functions but left opportunities for performance improvements. A number of VLSI design students at the University of Rochester have suggested various improvements in the design of the basic circuit elements.

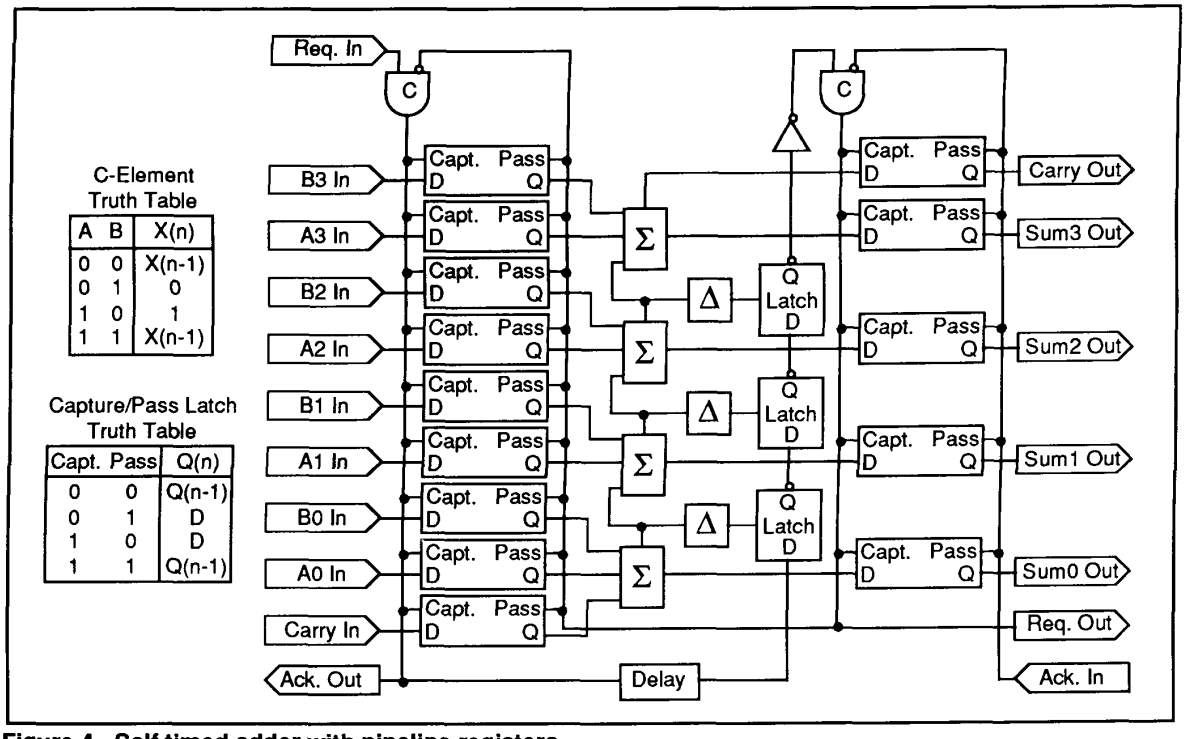

Figure 4 - Self timed adder with pipeline registers

Device	PMOS (μm)		NMOS (μm)	
	Width	Length	Width	Length
I1	18	2	9	2
I2	3	10	3	20
I3	6	2	3	2
I4	12	2	6	2
I5	6	2	3	2
I6	6	2	3	2
T1	6	2	3	2
T2	6	2	3	2
T3	6	2	3	2
T4	3	2	3	2

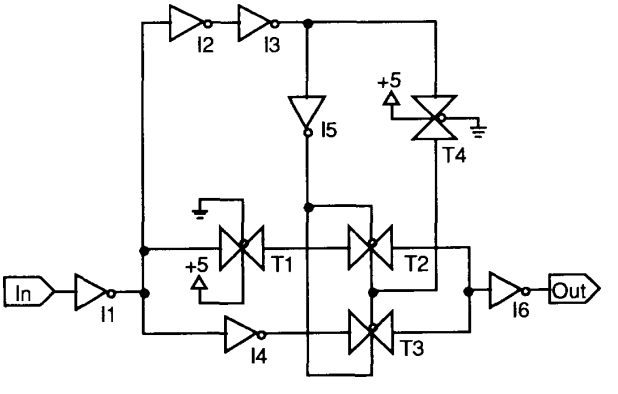

Figure 5 - Transition detector circuit

Table 1 - Performance summary

Description of Parameter	Time (ns)
Transition Detector (NMOS Length=20μm, PMOS Length=10μm)	
input high to output high propagation	2.1
input high to output low propagation	8.9
input low to high output pulse width	6.8
input low to output high propagation	2.4
input low to output low propagation	8.4
input high to low output pulse width	6.0
Inverting Latch	
input high to output low propagation (control low)	1.0
input low to output high propagation (control high)	1.9
Delay Element	
input high to output high propagation	8.5
input low to output low propagation	7.5
C Element	
input_b high to output low propagation (input_a low)	4.9
input_b low to output high propagation (input_a high)	5.4
Capture - Pass Latch	
input high to output high propagation (capture low, pass low)	3.0
input high to output high propagation (capture high, pass high)	3.0
pass low to output low propagation (capture low, input low)	2.3
pass high to output low propagation (capture high, input low)	1.6
One Bit Adder	
input_a high to sum_out low propagation (input_b low, carry_in high)	3.5
input_a high to carry_out high propagation (input_b low, carry_in high)	3.8
input_a low to sum_out high propagation (input_b low, carry_in high)	4.2
input_a low to carry_out low propagation (input_b low, carry_in high)	3.5
carry_in high to sum_out low propagation (input_a high, input_b low)	4.2
carry_in high to carry_out high propagation (input_a high, input_b low)	4.5
carry_in low to sum_out high propagation (input_a high, input_b low)	4.5
carry_in low to carry_out low propagation (input_a high, input_b low)	3.8

Much simpler C-element designs were proposed by the students, with most suggestion involving dynamic storage mechanisms; Susch [3] suggests a static design using a "weak" feedback inverter. Use of dynamic logic was also suggested in the design of the capture / pass latch and the transition detector controlled latch.

Singh [4] points out some opportunities for simplification of the carry portion of the adder by eliminating unnecessary transistors in transmission paths involving constant terms; there is no need for NMOS transistors in a transmission path which will only carry 5V, and there is no need for PMOS transistors in a transmission path which will only carry ground.

Both Susch and Singh suggest alternative transition detector approaches, but no significant performance gains or complexity reductions were obtained. The problems encountered in this area by these two students emphasizes the difficulty of designing a practical circuit which realizes this conceptually simple function.

Summary of performance of basic circuits

In the SPICE simulation of the devices, we used nominally sized transistors first, then did an initial Magic layout and used that preliminary layout to get a better idea of source and drain areas, perimeters, and squares. We added appropriately scaled inverters at the inputs and outputs of the devices in order to provide a test environment for each device which would be similar to the driving and load characteristics which it would encounter in the pipeline.

Based on the SPICE simulation, the performance of the basic circuits is summarized in Table 1.

Pipeline delay operation

The longest delay to carry output response to an input change is 4.5ns. The transition detector takes 2.4ns to respond. The time that is required from an input change to a transition detector response is thus 6.9ns. The transition detector stays high for 6.0ns minimum; it would seem not to cover the needed time for the next transition detector to respond. However, the latch adds additional delay in propagating the request signal so that by the time the request signal has arrived at the latch controlled by the next transition detector, it is blocked from proceeding.

Opportunities

In a ripple carry adder there are relatively few cases in which changing from one set of inputs to another will cause the carry to ripple throughout the adder. Nevertheless, this worst case resolution time determines the limits of performance of this adder in a synchronous system. In an asynchronous system such as we have outlined here, the adder becomes ready to process a new set of inputs as soon as the carries from the old inputs have settled down; on average, this will be some fraction of the worst case time. Hence, the average throughput of this adder in an asynchronous environment will be significantly better than the same adder in a synchronous

environment. This illustrates the power of self-timed asynchronous circuits: a circuit element which may be a bottleneck in a synchronous system based on its worst case response time may perform very well in an asynchronous system based on its average response time.

Although the basic concept of the transition detector / blocking latch method of detecting completion of a ripple carry addition appears to be validated by SPICE simulation, further opportunities for investigation are apparent.

Improving the performance of the scheme is a primary opportunity. Since each transition detector must block the completion signal for the amount of time required for the next carry and transition detector to respond, improvements in the response time of the carry and the transition detector will lead to a reduction in the completion time. Since the maximum time depends on the total number of carries in the adder, any small improvement in performance is multiplied by the number of bits in the adder. Improvements in other areas (the capture / pass latch, the C-element, etc.) are worthy of investigation, but don't hold the same potential for overall performance improvement.

Common layouts of alternative circuit implementations and consistent simulation methods would allow comparison of various design approaches. Simulation of assemblages of the basic circuit elements would demonstrate their performance in the actual pipeline environment. Based on the simulation results and estimates of the number and position of carries generated by streams of random input data, an estimate of pipeline throughput can be made.

The potential for using a transition detector / blocking latch scheme for monitoring the progress of other functions can be investigated. In particular, its application to wider adders and to multipliers needs to be explored.

References

1. David, I., Ginosar, R., and Yoeli, M., "An Efficient Implementation of Boolean Functions as Self Timed Circuits", Technical Report No. 678, Sept. 1988, Dept. Elect. Eng., Technion, Israel Institute of Technology, Haifa 32000, Israel.
2. Sutherland, I. E., "Micropipelines", Comm. of the ACM, June 1989, Volume 2, Number 6.
3. Susch, N., "Self-Timed 10 Bit Adder", Fall 1991 EE261 Project Report, Dept. of Elect. Eng., University of Rochester, Rochester, NY 14627.
4. Singh, H., Fall 1991 EE261 Project Report, Dept. of Elect. Eng., University of Rochester, Rochester, NY 14627.