

POWER OPTIMIZED MEMORY ORGANIZATION USING GATED DRIVER TREE

P. Sreenivasulu

K Srinivasa Rao

A. Vinaya Babu

Follow this and additional works at: <https://www.interscience.in/ijess>



Part of the [Electrical and Electronics Commons](#)

POWER OPTIMIZED MEMORY ORGANIZATION USING GATED DRIVER TREE

¹ P.Sreenivasulu, ² K.Srinivasa Rao, ³ A.Vinaya babu

¹Dr.S.G.I.E.T, MARKAPUR, Prakasam Dist., A.P, INDIA

² T.R.R College of Engineering, Inole, Patancheru, HYDERABAD, A.P, INDIA

³ JNTUH, Kukatpally, Hyderabad

Email: ¹patikineti.srinivas@gmail.com, ²principaltrr@gmail.com, ³dravinayababu@jntuh.ac.in

Abstract

This project presents circuit design of a low-power delay buffer. The proposed delay buffer uses several new techniques to reduce its power consumption. Since delay buffers are accessed sequentially, it adopts a ring-counter addressing scheme. In the ring counter, double-edge-triggered (DET) flip-flops are utilized to reduce the operating frequency by half and the C-element gated-clock strategy is proposed. A novel gated-clock-driver tree is then applied to further reduce the activity along the clock distribution network. Moreover, the gated-driver-tree idea is also employed in the input and output ports of the memory block to decrease their loading, thus saving even more power. And also, we are presenting less area overhead in this project by using FIFO (First In First Out) technique. FIFO is a technique, which is having the capability to store the DATA without any write operation and retrieving the DATA without any read operation.

I. INTRODUCTION

Portable multimedia and communication devices have experienced explosive growth recently. Longer battery life is one of the crucial factors in the widespread success of these products. As such, low-power circuit design for multimedia and wireless communication applications has been

processed, e.g., delay of one line of video signals, delay of signals within a fast Fourier transform (FFT) architectures [4], and delay of signals in a delay correlator [2]. Currently, most circuits adopt static random access memory (SRAM) plus some control/addressing logic to implement delay buffers.

In this paper, we propose to use double-edge-triggered (DET) flip-flops instead of traditional DFFs in the ring counter to halve the operating clock frequency. A novel approach using the C-elements instead of the

R-S flip-flops in the control logic for generating the clock-gating signals is adopted to avoid increasing the loading of the global clock signal. In the proposed new delay buffer, we use a tree hierarchy for the read/write circuitry of the memory module. For the write circuitry, in each level of the driver tree, only one driver along the path leading to the addressed memory word is activated. Similarly, a tree of multiplexers and gated drivers comprise the read circuitry for the proposed delay buffer. Simulation results show the effectiveness of the above

become very important. In many such products, delay buffers (line buffers, delay lines) make up a significant portion of their circuits [1]–[3]. Such serial access memory is needed in temporary storage of signals that are

techniques in power reduction. As an example, a 256 8 delay buffer chip is designed and fabricated. Measured results indicate its much better power performance than the same-size delay buffer based on existing commercial SRAM. The rest of this paper is organized as follows. Section II first introduces the conventional architecture for implementing delay buffers. Next, the proposed delay buffer using the new ring counter and gated driver trees for the read and write circuits of the memory module is described in Section III. Section IV then presents experimental results of the new delay buffer. Also, comparison in power and area of the new delay buffer with conventional SRAM-based delay buffers are given. Section V then concludes this paper.

II. CONVENTIONAL DELAY BUFFERS

The simplest way to implement a delay buffer is to use shift registers as shown in Fig. 1. If the buffer length is N and the word-length is b , then a total of Nb DFFs are required, and it can be quite large if a standard cell for

DFF is used. In addition, this approach can consume huge amount of power since on the average $Nb/2$ binary signals make transitions in every clock cycle. As a result, this implementation is usually used in short delay buffers, where area and power are of less concern.

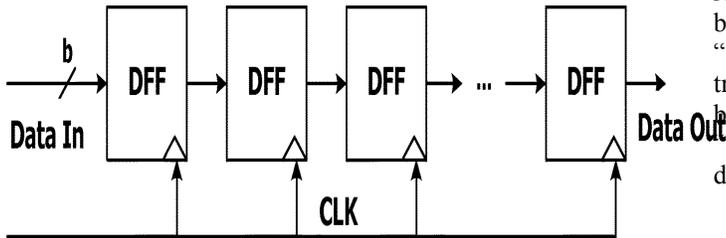


Fig.1. Delay buffer implemented by shift registers.

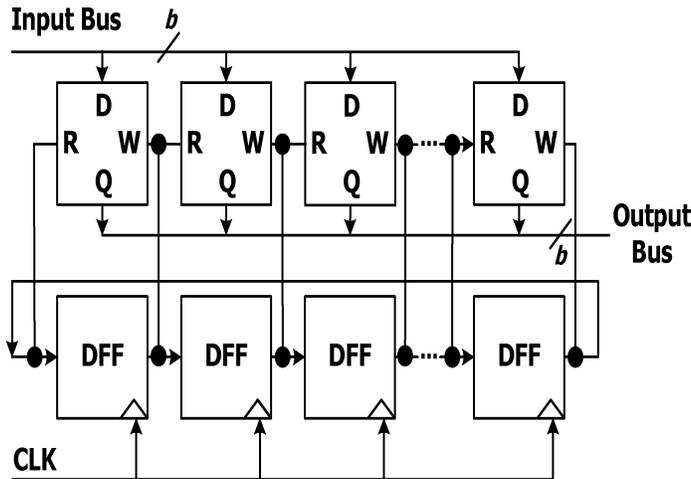


Fig. 2. Pointer-based delay buffer.

SRAM-based delay buffers are more popular in long delay buffers because of the compact SRAM cell size and small total area. Also, the power consumption is much less than shift registers because only two words are accessed in each clock cycle: one for write-in and the other for read-out. A binary counter can be used for address generation since the memory words are accessed sequentially. In fact, since the memory words are accessed sequentially, we can use a ring counter with only one rotating active cell to point to the words for write-in and read-out. This method, known as the pointer-based scheme [5], is illustrated in Fig. 2. The bottom row of D-type flip-flops is initialized with only one “1” (the active cell) and all the other DFFs are kept at “0.” When a clock edge triggers the DFFs, this “1”

signal is propagated forward. In their approach, every eight DFFs in the ring counter are grouped into one block. Then, a “gate” signal is computed for each block to gate the frequently toggled clock signal when the block can be inactive so that unnecessary power wasted in clock signal transitions is saved.

As shown in Fig. 3, when the input of the first DFF in a block is asserted, it sets the output of the R-S flip-flop to “1” at the next clock edge. Thus, the incoming “1” can be trapped in that block and continue to propagate inside the block. On the other hand, the successful propagation of “1” to the first DFF in the next block can henceforth shut down the unnecessary clock signal in the current block.

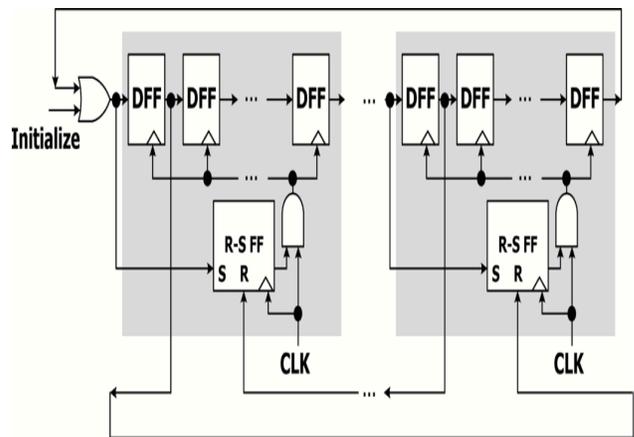


Fig. 3. Ring counter with clock gated by R-S flip-flop.

III. PROPOSED DELAY BUFFER

In the proposed delay buffer, several power reduction techniques are adopted. Mainly, these circuit techniques are designed with a view to decreasing the loading on high fan-out nets, e.g., clock and read/write ports.

A. Gated-Clock Ring Counter

Although some power is indeed saved by gating the clock signal in inactive blocks, the extra R-S flip-flops still serve as loading of the clock signal and demand more than necessary clock power. We propose to replace the R-S flip-flop by a C-element and to use tree-structured clock drivers with gating so as to greatly reduce the loading on active clock drivers. Additionally, DET flip-flops are used to reduce the clock rate to half and thus also reduce the power consumption on the clock signal. The proposed ring counter with hierarchical clock gating and the control logic is shown in Fig. 4. Each block contains one C-element to control

the delivery of the local clock signal “CLK ”to the DET flip-flops, and only the “CKE signals along the path passing the global clock source to the local clock signal are active. The “gate” signal (CKE) can also be derived from the output of the DET flip-flops in the ring counter.

The C-element is an essential element in asynchronous circuits for handshaking. One of its implementation is shown in Fig. 5(a) [7]. The logic of the C-element is given by $C^+ = AB + AC + BC$ where A as well as B are its two inputs C+ and as well C as are the next and current outputs. If $A=B$, then the next output C+ will be the same as A . Otherwise, $A \neq B$ and C+ remain unchanged. Since the output of C-element can only be changed when $A=B$, it can avoid the possibility of glitches, a crucial property for a clock gating signal. In order to reduce more power, we replace DFFs by double-edge-triggered flip-flops [8] [see Fig. 5(b)] and operate the ring counter at half speed .With such changes, the clock gating control mechanism in Fig. 4(a) is different from the one in Fig. 3. When the input of the last DET flip-flop in the at the output of the first DET flip-flop of block $jM/4i+1$. In a quad-tree driver architecture with four times more drivers in each level, all drivers need be

On the other hand, $2\log_4 M$ only drivers are activated in the worst case for the proposed gated-clock tree when two $\log_4 M$ drivers that are turned on, where $D=N/M$ is the number of DET flip-flops in one block. An example is illustrated in Fig. 6 with $M=64$. If the active “1” in the ring counter is propagated to the input of the last DET flip-flop,

previous block changes to “1” making both two inputs of the C-element the same, the clock signal in the current block will be turned on. When the output of the first DET flip-flop in the current block is asserted, then both inputs of the C-element in the previous block go to “0” and the clock for the previous block is disabled. In order to further diminish the loading on the global clock signal (“CLK”), we propose to use a driver tree distribution network for the global clock and activate only those drivers along the path from the clock source to the blocks that need to be driven by the clock. The “gate” signal for those drivers can be derived from the same clock gating signals of the blocks that they drive. Thus, in a quad-tree clock distribution network, the “gate” signal of the i th gate driver at the j th level (CKE) should be asserted when the active DET flip-flop (whose output is “1”) in the ring counter is inside the group of blocks with index from $(j-1)M/4i+1$ to $jM/4i$, where M is the number of blocks. To be precise, every clock gating signal will be on for two more cases, when “1” is at the input of last DET flip-flop in block $(j-1)M/4i$ and when “1” is

activated if no gating is applied and the number of active drivers is given by

$$M/4 + M/16 + \dots = M/3$$

drivers are activated in each of the $\log_4 M$ level. On the average, there are no more than $(1+(2/D$

$Q7B48$, then the clock enable signals, CKE_{2,13} and CKE_{1,4} are turned on. Subsequently, CKE_{2,13} and CKE_{1,4} can be delivered to block 49. On the contrary, when Q1B49 rises, it will turn off the clock enable

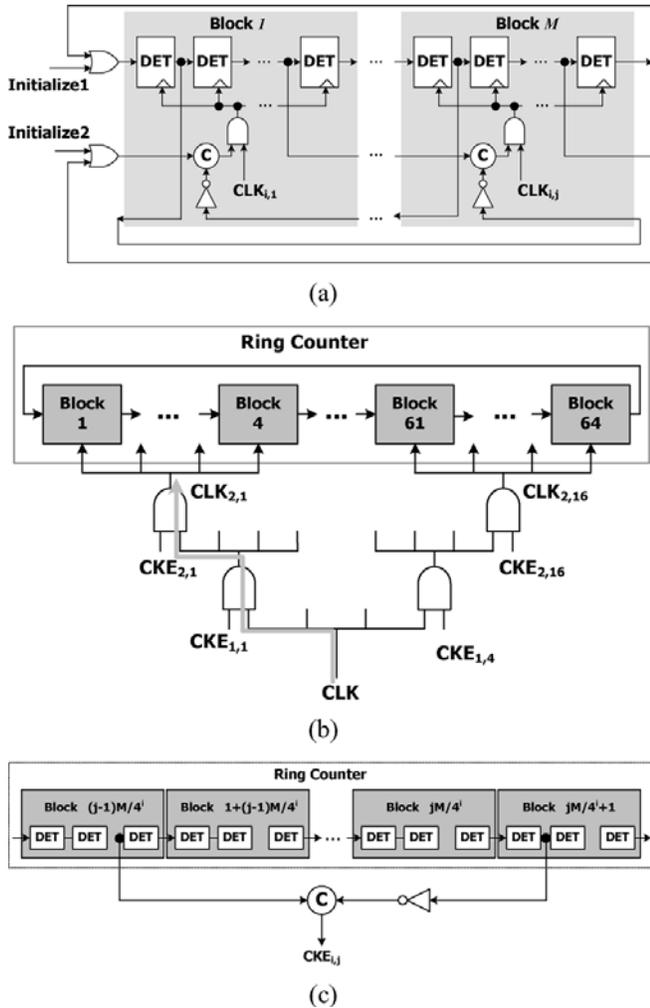


Fig. 4. (a) Ring counter with clock gated by C-elements, (b) tree-structured clock drivers with gating, and (c) control logic for clock enable signals.

signals of CKE_{2,12} and CKE_{1,3}. As a result, CLK_{2,12} and CLK_{1,3} stop driving their loading blocks. The loading of the clock signal in the proposed scheme can be analyzed as follows. Assume that a quad tree is used for clock drivers, then for a N length- ring counter constituted by a total of N flip-flops partitioned in M blocks.

- Loading of a traditional ring counter
 $N * LFF$
- Loading of [6]
 $D * LFF + M(LAND + LRS)$
- Loading of the proposed ring counter
 $D * LFF + 4(1 + (2/D)) \log_4 M * LAND$

where LFF, LAND, LRS and denote the loading of a D-type flip-flop clock input, an AND gate input and an R-S

flip-flop clock input, respectively. The simulation results indeed reflect the power consumption analysis in (3)–(5). It also indicates that it consumes less than 5% of the power of a same-length R-S flip-flop-based ring counter [6] (Fig. 3).

B. Gated-Driver Tree

To save area, the memory module of a delay buffer is often in the form of an SRAM array with input/output data bus as in [6]. Special read/write circuitry, such as a sense amplifier, is needed for fast and low-power operations. The memory words are also grouped into blocks. Each memory block associates with one DET flip-flop block in the proposed ring counter and one DET flip-flop output addresses a corresponding memory word for read-out and at the same time addresses the word that was read one-clock earlier for write-in. Fig. 7(a) depicts the tree-structured hierarchy of tri-state inverters used for delivering the input word to the addressed memory word. Note that only the driver tree for one input bit is shown. A b-bit delay buffer needs b sets of the circuitry in Fig. 7. The “enable” signal of the jth tri-state inverter at the ith level ($E_{i,j}$) should be asserted when the “1” is within one of the blocks from index $1+(j-1)M/4i$ to index $jM/4i$ in the ring counter, where M is the number of blocks and a quad tree is assumed. These signals are generated by a C-element and an inverter in a similar way as the circuit shown in Fig. 4(a) except that the input signal to the C-element from left (start-up) is the output of the first DET flip-flop in block $1+(j-1)M/4i$ and the signal from right (shut-off) is the output of the first DET flip-flop in block $1+jM/4i$. The corresponding timing diagram of control and data signals is shown in Fig. 7(b). As can be seen, the assertion of Q1B49, the output of the first DET flip-flop in block 49, activates E3.49, E.2,13, and E1,4 and simultaneously disables E3.48, E.2,12, and E1,3. The content of the memory word addressed by Q1B49, (address 384), is written by the input signal.

The loading of the input write circuitry can be estimated.

- Without gating strategy, it is
 $N * LLatch + 4/3(M-1) * LTri$
- With gated driver tree, it is
 $D * LLatch + 4 \log_4 M * LTri$

where LLatch and LTri are the loading of one latch and one tri-state buffer, respectively. In the estimation, a quad tree is assumed. We can see the logarithmical decrease in loading can dramatically reduce the power consumption. We have simulated input driver tree structures with and without the gating strategy by using 0.18- μ m CMOS technology and 1.8-V supply voltage at an operating frequency of 50 MHz. Due to the driving capability, buffer sizing is considered in the simulation

and equivalently extra loading of 12 LLatch must be included in (6) and (7) for the case $N=512$. If the loading ratio between LLatch and LTri is 1:2.8, the estimated power consumption ratio by (6) and (7) also matches very well with the simulated results.

The output sensing circuit has the same structure as that in Fig. 7(a) except that the signal flow direction is reversed. Data in the addressed memory element pass through several levels of

tri-state inverters that work as the multiplexer. The same set of enable signals (E) can be used to control these output tri-state inverters. Note that the proposed techniques can also be applied to variable-length delay buffers. First of all, the proposed ring counter can be made variable-length by including alternative signal paths selected by multiplexers to bypass some DET flipflops in the ring. Second, as all the control signals of the gated clock/driver tree are derived from the ring counter, a complete delay buffer with variable length can be constructed quite easily by including the latches and gated I/O driver trees. Of course, in this variable-length delay buffer, hardware corresponding to the maximum length must be implemented.

IV. EXPERIMENTAL RESULTS

A. Physical Design

A delay buffer based on the proposed techniques is designed and implemented in 0.18- μm CMOS technology. The standard 6-T SRAM cell is used in the delay buffer.

Eight DET flip-flops, eight memory words, and associated control logic are designed in a full-custom fashion and grouped as one block. We have simulated the proposed delay buffer with various lengths in 0.18 μm CMOS technology. The word-length is set to 8 bits. The area and power consumption are estimated from post layout simulation. In addition, we compared the simulated results with the values provided by a commercial SRAM compiler in the same technology. Since in each clock cycle, one read and one write operations are necessary for the delay buffer of length N , either one two-port SRAM with N words or two one-port SRAMs each with $N/2$ words is required.

A 256×8 delay buffer is designed and fabricated. In this circuit, a wordlength of one byte is adopted for this size is quite common in communication and video applications. Since each output of the ring counter is to activate one word line of the memory array, to achieve

compact area and high-speed operation, the outputs of the ring counter must be pitch-matched to the memory array and the ring counter is placed as close as possible to the memory cell. As shown in Fig. 9, to make the aspect ratio of the layout close to 1, the 128×8 delay buffer is folded into two 256×8 delay buffers. The core area of the 256×8 delay buffer is $0.232 \text{ m} \times 0.519 \text{ m}$. It is packaged in a 28-pin package, including eight output pins and twelve input pins.

B. Measurements

The measurement results are depicted in Fig. 10, where the maximum achievable clock rate under different supply voltages and the power consumption versus different supply voltages at the maximum operating frequency are shown. Table gives a brief summary of the low-power 256×8 delay buffer chip.

TABLE
COMPARISON OF MEASUREMENT RESULTS WITH
SRAM-BASED DELAY BUFFERS

	Dual-port SRAM Delay Buffer	Single-port SRAM Delay Buffer	This Work
Area	0.113 mm^2	0.062 mm^2	0.120 mm^2
Power	19913 μW	15359 μW	2556 μW

Finally, Table IV lists the comparison between the proposed chip and the delay buffers using commercial single-port as well as two-port SRAM. The proposed chip has a power consumption of about 17% of the single-port SRAM-based delay buffer, or 13% of the two-port SRAM-based delay buffer. The effectiveness of the proposed techniques in lowering power consumption is quite obvious.

C. Simulations for Scalability

To obtain a further insight about the scalability of the proposed delay buffer architecture in nanometer CMOS technology, we have run simulations of the proposed buffer with several different lengths in 90-nm and 65-nm CMOS technology.

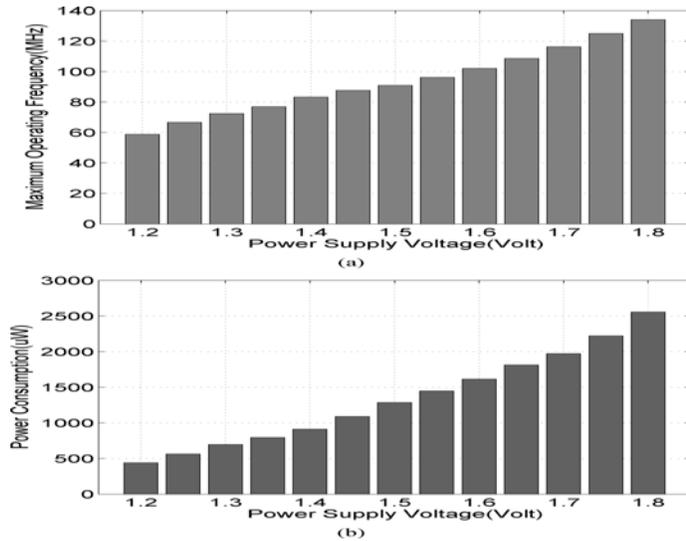


Fig. 10. Experimental results of (a) maximum clock rate and (b) power consumption at different supply voltage

in that the leakage power is almost negligible. Even in the more advanced 65-nm technology, the leakage power can be controlled to within an acceptable level for medium-length delay buffers with the dual-V_t approach. For longer-length delay buffers and for more advanced technology, other leakage reduction techniques such as the “sleep” transistors in SRAM (Latch) cells can help to reduce leakage power [9].

V. CONCLUSION

In this paper, we presented a low-power delay buffer architecture which adopts several novel techniques to reduce power consumption. The ring counter with clock gated by the C-elements can effectively eliminate the excessive data transition without increasing loading on the global clock signal. The gated-driver tree technique used for the clock distribution networks can eliminate the power wasted on drivers that need not be activated. Another gated-demultiplexer tree and a gated-multiplexer tree are used for the input and output driving circuitry to decrease the loading of the input and output data bus. All gating signals are easily generated by a C-element taking inputs from some DET flip-flop outputs of the ring counter. Measurement results indicate that the proposed architecture consumes only about 13% to 17% of the conventional SRAM-based delay buffers in 0.18- μ m CMOS technology. Further simulations also demonstrate its advantages in nanometer CMOS

technology. We believe that with more experienced layout techniques the cell size of the proposed delay buffer can be further reduced, making it very useful in all kinds of multimedia/communication signal processing ICs.

REFERENCES

- [1] W. Eberle et al., “80-Mb/s QPSK and 72-Mb/s 64-QAM flexible and scalable digital OFDM transceiver ASICs for wireless local area networks in the 5-GHz band,” *IEEE J. Solid-State Circuits*, vol. 36, no. 11, pp. 1829–1838, Nov. 2001.
- [2] M. L. Liou, P. H. Lin, C. J. Jan, S. C. Lin, and T. D. Chiueh, “Design of an OFDM baseband receiver with space diversity,” *IEE Proc. Commun.*, vol. 153, no. 6, pp. 894–900, Dec. 2006.
- [3] G. Pastuszak, “A high-performance architecture for embedded block coding in JPEG 2000,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 9, pp. 1182–1191, Sep. 2005.
- [4] W. Li and L. Wanhammar, “A pipeline FFT processor,” in *Proc. Workshop Signal Process. Syst. Design Implement.*, 1999, pp. 654–662.
- [5] E. K. Tsern and T. H. Meng, “A low-power video-rate pyramid VQ decoder,” *IEEE J. Solid-State Circuits*, vol. 31, no. 11, pp. 1789–1794, Nov. 1996.
- [6] N. Shibata, M. Watanabe, and Y. Tanabe, “A current-sensed high-speed and low-power first-in-first-out memory using a wordline/bitline-swapped dual-port SRAM cell,” *IEEE J. Solid-State Circuits*, vol. 37, no. 6, pp. 735–750, Jun. 2002.