

# AN OPTIMIZED SHIFT REGISTER USING PULSED LATCHES

**\*T.Shobakar, \*\*S.Muppudathi Sutha**

*\*PG Scholar, VLSI Design,*

*Sardar Raja College of Engineering, Alangulam, Tirunelveli, Tamilnadu*

*\*\*Assistant Professor, Department of ECE,*

*Sardar Raja College of Engineering, Alangulam, Tirunelveli, Tamilnadu*

## ABSTRACT

*Flip-flops (FFs) and latches are well known to be responsible for a large fraction of the power budget of microprocessors and VLSI systems. A new class of pulsed latches is introduced and experimentally assessed in 65-nm CMOS technology. Its conditional push-pull pulsed latch topology is based on a push-pull final stage driven by two split paths with a conditional pulse generator. A lowpower, high speed and area-efficient shift register using pulsed latches is designed in this project. A 256-bit shift register using pulsed latches was fabricated using a CMOS process with the power consumption is 1.2 mW at a 100 MHz clock frequency. Accordingly, the proposed class of pulsed latches goes beyond the current state of the art and is well suited for VLSI systems that require both high performance and energy efficiency. This is more than 1000 times faster than software processing running on a 3-GHz general-purpose processor.*

## INTRODUCTION

Very-large-scale integration (VLSI) is the process of creating an integrated circuit (IC) by combining thousands of transistors into a single chip. VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device. Before the introduction of VLSI technology most ICs had a limited set of functions it could perform. An electronic circuit might consist of a CPU, ROM, RAM and other glue logic. VLSI lets IC makers add all of these into one chip. Structured VLSI design is a modular methodology originated by Carver Mead and Lynn Conway for saving microchip area by minimizing the interconnect fabrics area. This is obtained by repetitive arrangement of rectangular macro blocks which can be interconnected using wiring by abutment. An example is partitioning the layout of an adder into a row of equal bit slices cells. In complex designs this structuring may be achieved by hierarchical nesting. VLSI Design presents state-of-the-art papers in VLSI design, computer-aided design, design analysis, design implementation, simulation and testing. Its scope also includes papers that address technical trends, pressing issues, and educational aspects in VLSI Design.

In digital circuits, a shift register is a cascade of flip flops, sharing the same clock, in which the output of each flip-flop is connected to the "data" input of the next flip-flop in the chain, resulting in a circuit that shifts by one position the "bit array" stored in it, shifting

in the data present at its input and shifting out the last bit in the array, at each transition of the clock input. More generally, a shift register may be multidimensional, such that its "data in" and stage outputs are themselves bit arrays: this is implemented simply by running several shift registers of the same bit-length in parallel.

Shift registers can have both parallel and serial inputs and outputs. These are often configured as 'serial-in, parallel-out' (SIPO) or as 'parallel-in, serial-out' (PISO). There are also types that have both serial and parallel input and types with serial and parallel output. There are also 'bidirectional' shift registers which allow shifting in both directions: L→R or R→L. The serial input and last output of a shift register can also be connected to create a 'circular shift register'.

Shift register IC's are generally provided with a clear or reset connection so that they can be "SET" or "RESET" as required. Generally, shift registers operate in one of four different modes with the basic movement of data through a shift register being:

- Serial-in to Parallel-out (SIPO) - the register is loaded with serial data, one bit at a time, with the stored data being available at the output in parallel form.
- Serial-in to Serial-out (SISO) - the data is shifted serially "IN" and "OUT" of the register, one bit at a time in either a left or right direction under clock control.
- Parallel-in to Serial-out (PISO) - the parallel data is loaded into the register simultaneously and is shifted out of the register serially one bit at a time under clock control.

## EXISTING SYSTEM

The clock phase generator can be shared among multiple latches to amortize its overhead. It is useful to observe that the width of  $CP_f$  and  $CP_r$  pulses determines the width of the transparency window of  $CP^3L$  latch in which the input can affect the output. From a design point of view, the width of the transparency window can be modified by changing the delay of the inverters within the clock phase generator in Fig. 3.5. The effect of process variations on timing can be compensated through post-silicon tuning of the pulse width, possibly sharing the tuning circuitry among multiple latches [1], [20], [21]. In this, no tuneability is added to the considered pulsed latches since the addition of such feature would impact area/energy of any pulsed latch equally. Indeed, almost all existing pulsed latches adopt the same pulse generator topology. The delay stage in the feedback path in Figs. 3.3–3.5 generates a delayed replica QD of the output Q, and is implemented by the two inverters M13–M14 and M25–M26 in Fig. 3.5. Actually, only slow transistors M25–M26 are added to implement such delay, as the inverter M13–M14 is already available (i.e., M13–M14 are used to both latch and delay the output). This delay stage makes sure that QD is kept stable at its previous value during the transparency window, thereby preventing glitches in  $CP_r$  and  $CP_f$  and reducing dynamic energy, as discussed in the following. Without the delay stage, the output Q would be connected directly to the pseudo-NAND/NOR in Fig. 3.5, hence any

output transition within the transparency window immediately triggers the generation of an additional (undesired) pulse.

As shown in detail in Fig. 3.7, which refers to the case where  $Q$  is directly connected to the pseudo-NAND/NOR, a falling transition of  $Q$  following the same input transition immediately triggers a high pulse in  $CP_r$ , as the pseudo-NOR in Fig. 3.5 temporarily has all pMOS transistors M22–M24 ON during the transparency window (i.e., the  $CP_r$  time slot in Fig. 3.6).

### 3.3.2. CSP<sup>3</sup>L: Conditional Shareable Push–Pull Pulsed Latch

In CP<sup>3</sup>L, the pulse generator cannot be shared among multiple latches since pseudo-NOR/NAND are driven by  $QD$ , which is different for each latch. In this subsection, we present a different implementation of the same concept by integrating the conditional logic in the latch so that the whole pulse generator can be shared. In CSP<sup>3</sup>L, static NAND/NOR gates are introduced in the shareable pulse generator to generate the pulses  $CP_{f,ext}$  and  $CP_{r,ext}$  that are distributed to multiple latches and have the same role as  $CP_f$  and  $CP_r$  had in CP<sup>3</sup>L. In each latch, such external pulses are enabled through the switches implemented by M16–M22, which implement the conditional pulse selection logic. The latter comprises two transmission gates and two small keepers to maintain the same operation as before. As discussed above, the delay stage M23–M26 is introduced in the feedback path (two more than CP<sup>3</sup>L since the transmission gates need complementary control signals). The resulting transistor count is the same as CP<sup>3</sup>L, hence CSP<sup>3</sup>L area is expected to be roughly the same as CP<sup>3</sup>L (excluding the shareable part). Since CSP<sup>3</sup>L is based on the same concept as CP<sup>3</sup>L, operation is very similar. The main difference is in the conditional pulse selection logic, which enables the propagation of either  $CP_{f,ext}$  or  $CP_{r,ext}$  to the half latches, according to the value of the delayed output replica  $QD$ . In particular, if  $QD = 1$  ( $QD = 0$ ) the fall (rise) path is activated, as the transmission gate M15–M16 (M19–M20) transfers the  $CP_{f,ext}$  ( $CP_{r,ext}$ ) pulse to the input of the half latch M1–M3 (M4–M6), similar to the pseudo-NAND (pseudo-NOR) of CP<sup>3</sup>L in Fig. 3.5. As a minor difference from CP<sup>3</sup>L, the input capacitance seen from  $CP_{f,ext}$  and  $CP_{r,ext}$  in CSP<sup>3</sup>L depends on  $Q$ , which may lead to data-dependent clock skew. In practical cases, this is not a concern considering that pulsed latches inherently tolerate a significant amount of skew.

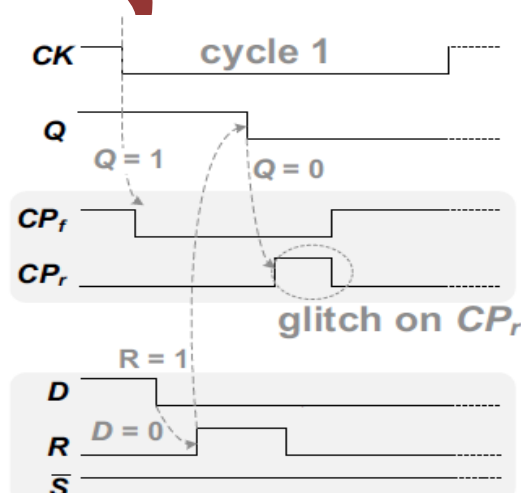
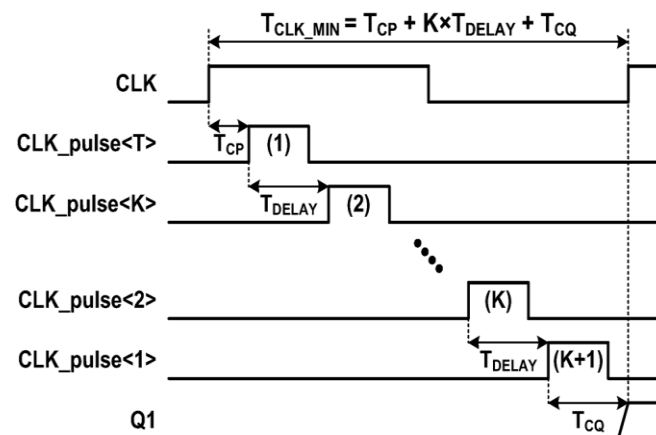


Fig. 1. Glitch in  $CP_r$  occurring if no delay stage is inserted in the feedback

## PROPOSED SYSTEM

The power optimization is similar to the area optimization. The power is consumed mainly in latches and clock-pulse circuits. Each latch consumes power for data transition and clock loading. When the circuit powers are normalized with a latch, the power consumption of a latch and a clock-pulse circuit are 1 and  $\alpha_p$ , respectively. The total power consumption is also  $\alpha_p \times (K + 1) + N(1 + 1/K)$

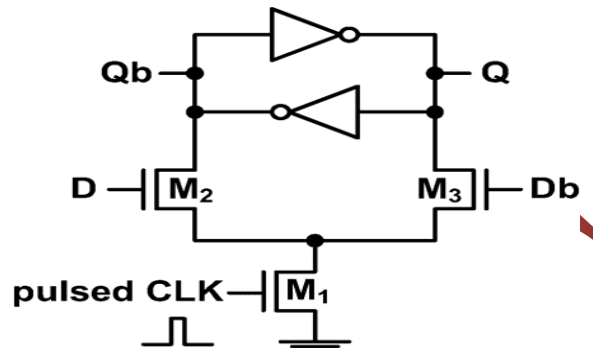


**Fig. 4.7. Minimum clock cycle time of the proposed shift register**

An integer  $K$  for the minimum power is selected as a divisor of  $N$ , which is nearest to  $\sqrt{N/\alpha_p}$ . In selection, the clock buffers in Fig. 4.6 are not considered. The total size of the clock buffers is determined by the total clock loading of latches. Although the number of latches increases from  $N$  to  $N(1+1/K)$ , the increment ratio of the clock buffers is small. The number of clock buffers is  $K$ . As  $K$  increases, the size of a clock buffer decreases in proportion to  $1/K$  because the number of latches connected to a clock buffer ( $M=N/K$ ) is proportional to  $1/K$ . Therefore, the total size of the clock buffers increases slightly with increasing  $K$  and the effect of the clock buffers can be neglected for choosing  $K$ .

The maximum number of  $K$  is limited to the target clock frequency. As shown in Fig. 4.7 the minimum clock cycle time ( $T_{CLK\_MIN}$ ) is  $T_{CP} + K \times T_{DELAY} + T_{CQ}$ , where  $T_{CP}$  is the delay from the rising edge of the main clock signal (CLK) to the rising edge of the first pulsed clock signal (CLK\_pulse<T>),  $T_{DELAY}$  is the delay of two neighbor pulsed clock signals, is the delay from the rising edge of the last pulsed clock signal (CLK\_pulse<1>) to the output signal of the latch Q1.  $T_{CLK\_MIN}$  is proportional to  $K$ . As  $K$  increases, the maximum clock frequency ( $f_{CLK\_MAX} = 1/T_{CLK\_MIN}$ ) decreases in proportion to  $1/K$ . Therefore, must be selected under the maximum number which is determined by the maximum clock frequency of the target applications. The  $K+1$  pulsed clock signals in Fig. 4.7 are supplied to all sub shift registers. Each pulsed clock signal arrives at the sub shift registers at different time due to the pulse skew in the wire. The pulse skew increases proportional to the wire distance from the delayed pulsed clock generator. All pulsed clock signals have almost the same pulse skews when they arrive at the same sub shift register. Therefore, in the same sub shift register, the pulse skew differences between the pulsed clock signals are very small. The clock pulse intervals larger

than the pulse skew differences cancel out the effects of the pulse skew differences. Also, the pulse skew differences between the different sub shift registers do not cause any timing problem, because two latches connecting two sub shift registers use the first and last pulsed clocks (CLK\_pulse(T) and CLK\_pulse(1)) which have a long clock pulse interval



**Fig. 2. Schematic of the SSASPL**

In a long shift register, a short clock pulse cannot through a long wire due to parasitic capacitance and resistance. At the end of the wire, the clock pulse shape is degraded because the rising and falling times of the clock pulse increase due to the wire delay. A simple solution is to increase the clock pulse width for keeping the clock pulse shape. But this decreases the maximum clock frequency. Another solution is to insert clock buffers and clock trees to send the short clock pulse with a small wire delay. But this increases the area and power overhead. Moreover, the multiple clock pulses make the more overhead for multiple clock buffers and clock trees.

## RESULTS AND DISCUSSION

This paper presents new techniques to evaluate the energy and delay of flip-flop and latch designs and shows that no single existing design performs well across the wide range of operating regimes present in complex systems. We propose the use of a selection of flip-flop and latch designs, each tuned for different activation patterns and speed requirements. We illustrate our technique on a pipelined MIPS processor datapath running SPECint95 benchmarks, where we reduce total flip-flop and latch energy by over 60% without increasing cycle time Pulsed latch structures employ an edge-triggered pulse generator to provide a short transparency window. Compared to master-slave flip-flops, pulsed latches have the advantages of requiring only one latch stage per clock cycle and of allowing time-borrowing across cycle boundaries. The major disadvantages of pulsed latch structures are the increased susceptibility to timing hazards and the energy dissipation of the local clock pulse generators. Pulse generators can be shared among a few latch cells to reduce energy, if care is taken that the pulse shape does not degrade due to wire delay, signal coupling and noise. We measured designs both with individual pulse generators and with pulse generators shared among four latch bits, in which case we divide the pulse generator energy among the four latch instances.



## CONCLUSION

This project proposes a low-power and area-efficient shift register using pulsed latches. The shift register reduces area and power consumption by replacing flip-flops with pulsed latches. The timing problem between pulsed latches is solved using multiple non-overlap delayed pulsed clock signals instead of a single pulsed clock signal. A small number of the pulsed clock signals is used by grouping the latches to several sub shifter registers and using additional temporary storage latches. The proposed shift register saves 37% area and 44% power compared to the conventional shift register with flip-flops.

## REFERENCES

- [1] P. Reyes, P. Reviriego, J. A. Maestro, and O. Ruano, "New protection techniques against SEUs for moving average filters in a radiation environment," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 4, pp. 957–964, Aug. 2007.
- [2] M. Hatamian et al., "Design considerations for gigabit ethernet 1000 base-T twisted pair transceivers," *Proc. IEEE Custom Integr. Circuits Conf.*, pp. 335–342, 1998.
- [3] H. Yamasaki and T. Shibata, "A real-time image-feature-extraction and vector-generation vlsi employing arrayed-shift-register architecture," *IEEE J. Solid-State Circuits*, vol. 42, no. 9, pp. 2046–2053, Sep. 2007.
- [4] H.-S. Kim, J.-H. Yang, S.-H. Park, S.-T. Ryu, and G.-H. Cho, "A 10-bit column-driver IC with parasitic-insensitive iterative charge-sharing based capacitor-string interpolation for mobile active-matrix LCDs," *IEEE J. Solid-State Circuits*, vol. 49, no. 3, pp. 766–782, Mar. 2014.
- [5] S.-H. W. Chiang and S. Kleinfelder, "Scaling and design of a 16-megapixel CMOS image sensor for electron microscopy," in *Proc. IEEE Nucl. Sci. Symp. Conf. Record (NSS/MIC)*, 2009, pp. 1249–1256.
- [6] S. Heo, R. Krashinsky, and K. Asanovic, "Activity-sensitive flip-flop and latch selection for reduced energy," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 9, pp. 1060–1064, Sep. 2007.
- [7] S. Naffziger and G. Hammond, "The implementation of the next generation 64 b titanium microprocessor," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2002, pp. 276–504.
- [8] H. Partovi et al., "Flow-through latch and edge-triggered flip-flop hybrid elements," *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, pp. 138–139, Feb. 1996.
- [9] E. Consoli, M. Alioto, G. Palumbo, and J. Rabaey, "Conditional push-pull pulsed latch with 726 fJops energy delay product in 65 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2012, pp. 482–483.
- [10] V. Stojanovic and V. Oklobdzija, "Comparative analysis of masterslave latches and flip-flops for high-performance and low-power systems," *IEEE J. Solid-State Circuits*, vol. 34, no. 4, pp. 536–548, Apr. 1999.