Logic Design for Low-Voltage / Low-Power CMOS Circuits

C. Piguet, J-M. Masgonty, V. von Kaenel, T. Schneider

CSEM Centre Suisse d'Electronique et de Microtechnique SA
Maladière 71, 2000 Neuchâtel, Switzerland

Abstract

The reduction of the supply voltage of integrated circuits is very efficient to save power. Logic design at low supply voltage has to be performed while considering the critical path. Logic modules on the critical path have to be decomposed and parallelized in a different way than those which are not on the critical path. Parallelization of logic circuits results in the reduction of the operating frequency as well as the supply voltage. Consequently, the power consumption is reduced of several factors while maintaining the same throughput provided by the non parallelized structures. Several examples are described such as parallelized synchronous counters and shift registers.

1. Introduction

As the power consumption is a major issue, designers are required to design low-voltage circuits without any speed loss. New logic design methodologies have to be used if the V_Ts are not decreased [1, 2] at the same extent. However, different approaches have to be used for logic modules/gates located on, or not located on, the critical path. At the gate level, decomposition of complex gates on the critical path results in faster gates that can be supplied at lower supply voltages [3]. Parallelization of modules on the critical path has been proposed to compensate for the speed reduction at low supply voltages [4, 5, 6]. However, parallelization generally results in the duplication of execution units and could be considered as expensive by designers. Some examples described in this paper do not present such a drawback. Other design methods have to be used for the logic modules/gates that are not on the critical path, such as simplicity, capacitance, activity and frequency reduction.

2. Critical path

At a reduced supply voltage, the most important problem is to compensate for the speed loss of the logic modules and gates that are placed on the critical path. However, in a random logic block, some logic is not on the critical path. Such a logic could be supplied at a lower supply voltage without any modification, or could be modified in order to save power, i.e. by adopting simpler architecture or gate design and smaller transistors.

Each logic module could be supplied by its own Vdd in order to provide the right supply voltage for the required speed. The largest power saving is reached if Vdd is such as the module works as close as possible of its frequency limit. However, such an architecture would require many DC-DC converters and level shifters, and could seem today unrealistic. Another approach that will be described in this paper is a common low supply voltage for all the logic modules. It requires new design and logic synthesis of the module architectures in such a way that all the modules are close to their frequency limit at the same common low Vdd.

3. Cell Library Categories

The choice of very high-speed cells placed on the critical path is obvious. However, cells that are not placed on the critical path contain oversized transistors, with a higher parasitic capacitance. This increases the power consumption and slows down even the cells which are on the critical path. Figure 1 shows a simple example in which a very fast cell is loaded with many other cells that are not on the critical path. If these cells contain oversized transistors, the load capacitance of the very fast cell is increased, resulting in a decreased speed. On the contrary, if the other cells are low-power cells with small or unsized transistors, the speed of the first cell will be higher and the power consumption reduced.

Several cell categories are therefore useful in standard cell libraries, for instance three categories in [3,7] : a low-power category with unsized transistors, a high-speed category, and very high-speed cells for which the maximum number of transistors in series is limited to two. If more categories are needed, an automatic transistor sizing procedure [8] has to be used.

4. Complex Gate Decomposition

Two different cases have to be considered : if such a complex gate is on the critical path, it has to be decomposed as its delay is reduced and it can fit to a reduced Vdd [3]. However, if such a complex gate is not on the critical path, it can work as such at a reduced Vdd, and the decomposition results in more transistors and in generally more parasitic capacitance.

According to a rough gate delay model, a N-input gate (for instance a NAND) contains a branch with N transistors in series, resulting in an increased internal delay of $N*\partial$. Furthermore, the internal parasitic capacitance is also roughly increased of a factor N. The internal delay of a N-input gate is therefore $N^2*\partial$. The load delay of an N-input gate is $N*\Delta$ as the output capacitance has to be charged or discharged by a branch with N transistors in series. The total delay of an N-input gate is therefore delay = $N^2*\partial + N*\Delta$.

For a 6-input NAND gate (Fig. 2), the total delay is $\approx 36\partial + 6\Delta$. If such a 6-input gate is decomposed (Fig. 2), its critical path is made of 3 simple gates in series (a 3-input gate, a 2-input gate, an inverter), resulting in a shorter total delay of $14\partial + \Delta$. Less parasitic capacitance is switched from the input to the output. The supply voltage can therefore be reduced to save power. However, if such a decomposed gate is supplied at the original Vdd, no power could be saved as a simple gate could switch without an output transition (Fig. 2). If a complex gate is not on the critical path, it works at the reduced Vdd. Gate decomposition is therefore not necessary to fit to the speed requirements.

Such a gate decomposition has been proposed many years ago [9] for hierarchical bus drivers. Many tri-state gates on the same bus result in a very complex CMOS gate and a very large bus capacitance (Fig. 3). A hierarchy of tri-state gates or a tree of 2:1 multiplexers is a much better approach (Fig. 4) for low voltage circuits.

5. Low-Voltage Circuit Parallelization
Circuit parallelization has been proposed to maintain, at a reduced Vdd, the throughput of logic modules that are placed on the critical path [4, 5, 6]. This can be achieved with M parallel units clocked at f/M. Results are provided at the nominal frequency f through an output multiplexer controlled at f. Each unit can compute its result in a time slot M times longer, and can therefore be supplied at a reduced supply voltage. If the units are datapaths or processors [4], they have to be duplicated, resulting in an M times area and switched capacitance increase. Applying the well-known power formula, one can write the following :

$$P = M*C * f/M * Vdd^2 = C * f * Vdd^2$$

One could deduce that power is saved only if Vdd is reduced. However, as operating frequency is reduced, the use of cells with smaller or unsized transistors results in a power reduction.

Furthermore, some parallelized logic modules do not require a M-unit duplication. This is the case, for instance, for memories, in which each unit contains 1/M data or instructions, resulting in the same total area to store the information and in the same C or smaller C' total switched capacitance if cells with unsized transistors are used (Fig. 5). In such a case, the power is the following :

$$P = C' * f/M * Vdd^2$$

At first order, power could be saved even if Vdd is not reduced. However, some overhead has to be considered, such as the address registers duplication and the output multiplexer (Fig. 5). If this overhead is not too expensive, such a parallelization scheme has to be considered for logic modules that are not on the critical path. At a low Vdd, they are working without parallelization. At the same low Vdd, power could be saved if they are parallelized at the cost of a small overhead. Memories, shift registers, serial-parallel converters, provide interesting examples.

In parallelized modules, operations of the execution units or data accesses in memories are performed in an overlapped or interleaved fashion (Fig. 6). The result is therefore provided with a M-1 latency delay compared to a non parallel architecture. The operation or access of a unit 2 is started before the completion of the operation of unit 1. Therefore, M successive computations have not to be dependent on each other. Controllers with a fixed sequence of commands without any branch instruction or RAM memories used to store coefficients for programmable FIR filters can be designed according to the structure of Fig. 5. One can see on the timing diagram of Fig. 6 that the output multiplexer can be controlled at $f/2$.

Generally, operations are dependent on each other. The most obvious way to solve such a problem is to insert a delay, i.e. a load delay for dependent operations and a branch delay for branch instructions. However, for some particular cases in which the operation dependency is determined, such as synchronous counters, parallelization can be performed without delays although the next state is dependent on the previous state.

6. Parallelized Synchronous Counter
Figure 7 depicts the flow table, the flow graph as well as the FSM structure of a 3-bit Gray code synchronous counter. The parallelization is achieved with two state machines that implement half the states (Fig. 8) of the original flow graph (Fig. 7). Each sub-graph contains one over two successive states of the original flow graph [10]. This results in a reduction of the number of transistors in the combinational circuits (Fig. 9). However, the total number of transistors is increased of about a factor 2, as the

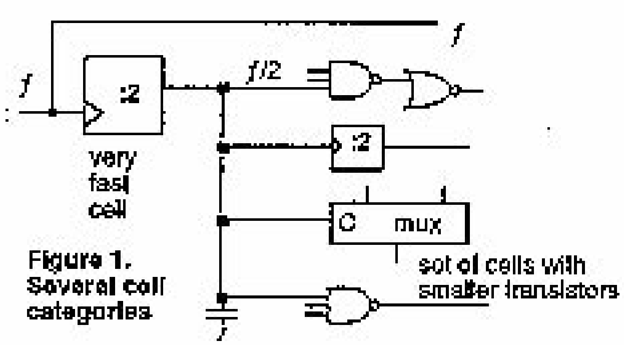

Figure 1. Several cell categories

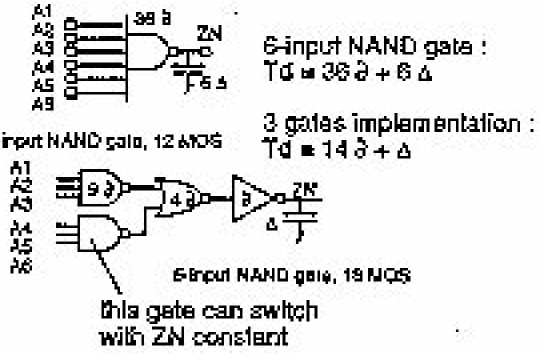

Figure 2. Complex Gate on the critical path

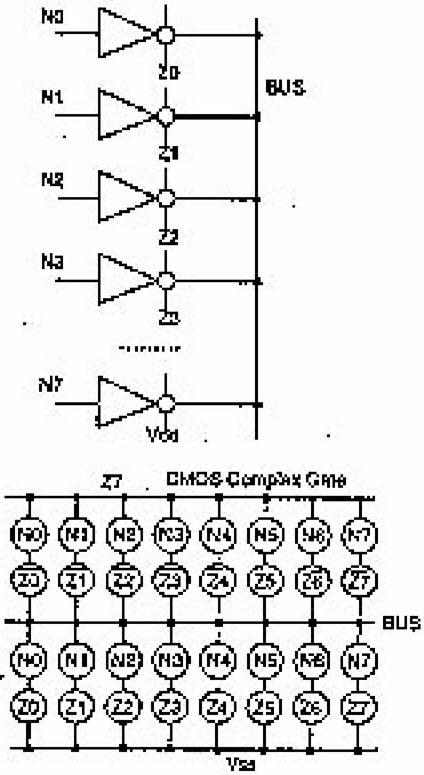

Figure 3. BUS Driver

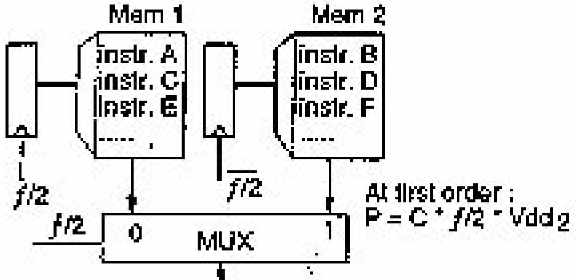

Figure 5. Memory Parallelization

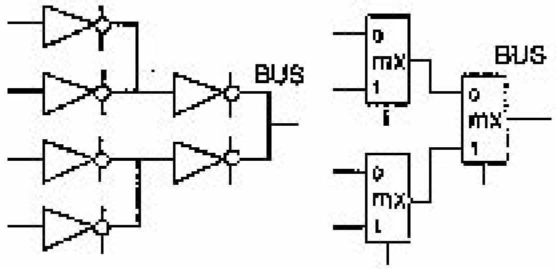

Figure 4. Hierarchical BUS drivers

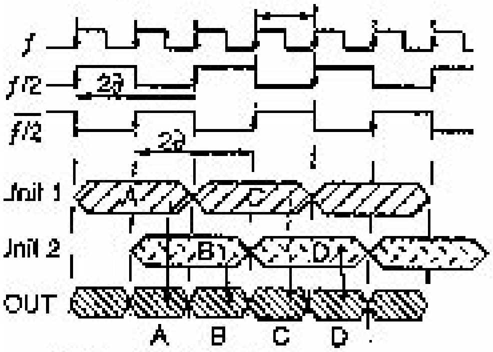

Figure 6. Timing Diagram

ABC	A+B+C+
000	001
001	011
010	111
011	101
100	100
101	110
110	010
111	000

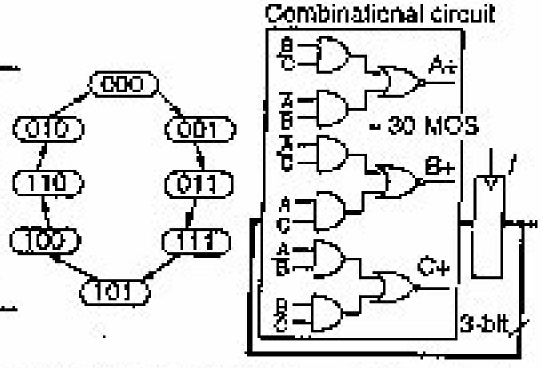

Figure 7. three-bit Counter Flow Table and Flow Graph

ABC	A+B+C+
000	011
011	101
101	110
110	000

Two parallel flow tables

ABC	A+B+C+
001	111
111	100
100	010
010	001

One over two successive states belongs to another flow table

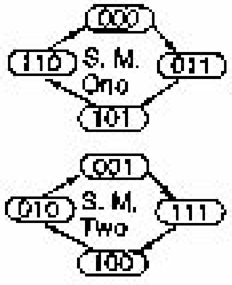

Figure 8. Parallelized Counter

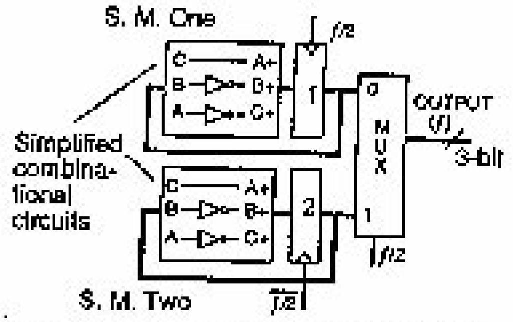

Figure 9. Parallelized Counter Structure

registers are duplicated. One could deduce that the switched capacitance C is increased of a factor 2, but smaller transistors can be used as the registers are clocked at $f/2$. The result could be, at first order, a non increased switched capacitance.

Speed comparison can be performed with a rough delay model in which the latches of the D-Flip-Flops and the output multiplexer present the delay Δ while the combinational circuits present a delay $X*\Delta$.

In the non parallelized counter (Fig. 10), the total delay is :

$$T= 2*\Delta+X*\Delta.$$

For several values of X, the maximum frequency $f$max is plotted in Figure 11. The parallelized counter (Fig. 9) working at $f/2$ has therefore a total delay (Fig. 10):

$$2*T=2*\Delta+X*\Delta,$$

$$\text{i.e.} \qquad T=(1+0.5*X)*\Delta$$

The corresponding maximum frequency $f$max is also plotted in Figure 11 (note that only $f/2$ is physically applied to the counter). However, the maximum frequency is limited by another condition which is the sum of the delay Δ of the slave of the D-Flip-Flop and the delay Δ of the output multiplexer (Fig. 10):

$$T=2*\Delta$$

Such a relation is plotted as an horizontal line that limits the maximum frequency f when X is smaller than 2 (Fig. 11). This means that the maximum frequency cannot be increased of a factor M in a M-parallelization scheme if the delay of the output multiplexer is equal or larger than the delay of the combinational circuits of the considered states machines.

Performances can be compared with a non parallelized counter with a power consumption $P=F*C*Vdd^2$ and N operations/sec. In a first case, at the same Vdd, the throughput of the parallel counter (Fig. 9) is increased at 2*N with a maximum frequency $f=2*F/2=F$. Transistor sizes cannot be decreased, resulting in a capacitance 2*C and a power=2*P. A faster counter is obtained. In a second case, the same throughput N is achieved at a frequency $f=F/2$ at the same Vdd. Smaller transistors can be used, resulting in a reduced power P' at the same Vdd due to the capacitance C' reduction. Finally, at a reduced Vdd", the same throughput N is achieved with a reduced frequency $f=F/2$. Transistors sizes cannot be reduced due to the Vdd reduction. With a switched capacitance 2*C, the power P" reduction is only due to the Vdd reduction.

Figure 12 shows the parallelization of a program counter with branch instructions. Registers are realized with one master clocked at f and two slave latches clocked at $f/2$. The interleaved incrementers by 2 also work at $f/2$. For a branch instruction, the output of the incrementers are not used. A first branch address is loaded in the master during the jump instruction as well as the address+1 in the next clock period. Therefore, the memory contains in the same jump instruction two addresses A and A+1. The overhead due to the parallelization is quite expensive. The master, the multiplexer and the memory have to work at the frequency f.

Figure 13 shows the parallelization of an up-down counter. The same phased register structure is applied. To be capable of switching immediately up and down while the input X is modified, the output multiplexer has 6 different inputs. In the up mode, interleaved +2 incrementers are used while -2 are used in the down mode. However, during the transition between the two modes, the direct counter state has to be used. The parallelization overhead is also quite high. These examples show that the parallelization with branch operations are quite expensive.

7. Parallelized Shift Register

Figure 14 shows the structure of a parallelized shift register. Such structures have been proposed for CCD memories and parallel pipelines in computers [14, 15]. The input is successively provided to the upper or to the lower half shift register at a reduced frequency, while the output multiplexer restores the output at the frequency f. There is no latency, as the combinational circuit of the state machine "shift register" is implemented by simple wires, resulting in no associated delay (X=0 in Fig 11). The total number of D-flip-flops is the same as for the non parallelized shift register.

For the non parallelized shift register, the maximum frequency is limited by the delays of the latches of the D-flip-flop, i.e. $T=2*\Delta$ (Fig. 10 with X=0). For the parallelized shift register, the maximum frequency is limited by one latch delay and the output multiplexer delay, i.e. $T=2*\Delta$. As shown in Figure 11, for X=0, the maximum frequency of the parallelized structure is the same as for the classic structure (a $f$max=100 MHz classic shift register can be replaced by a $f/2$=50 MHz parallelized shift register, but it is impossible to increase $f/2$ > 50 MHz). Such a parallelization does not provide faster shift registers. It is therefore impossible to reduce Vdd if the shift register in on the critical path. However, at the same Vdd, parallelized shift registers with the same throughput present a reduced power consumption

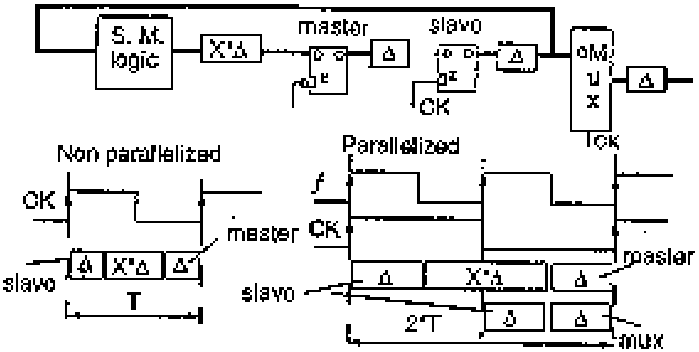

Figure 10. Delay Model

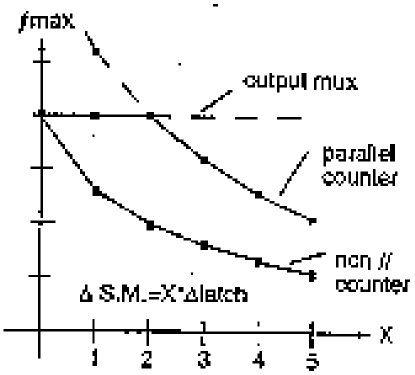

Fig.11 fmax limits

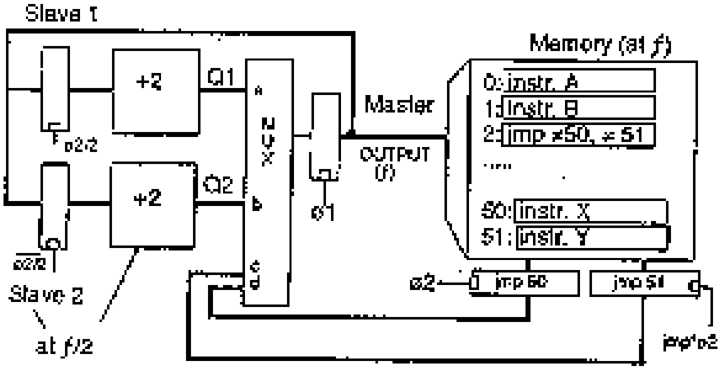

Figure 12. Parallelized Program Counter

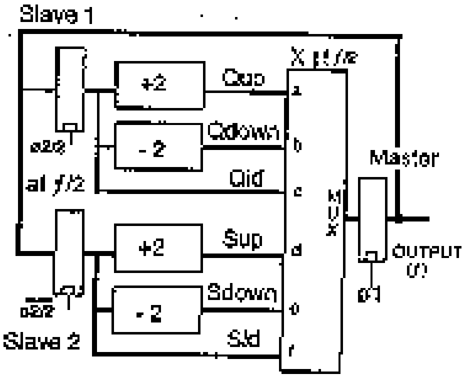

Figure 13. Up-Down Counter Parallelization

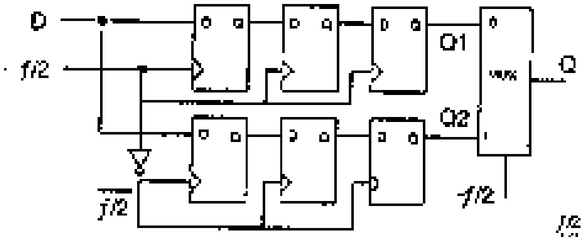

Figure 21 Parallelized Shift register

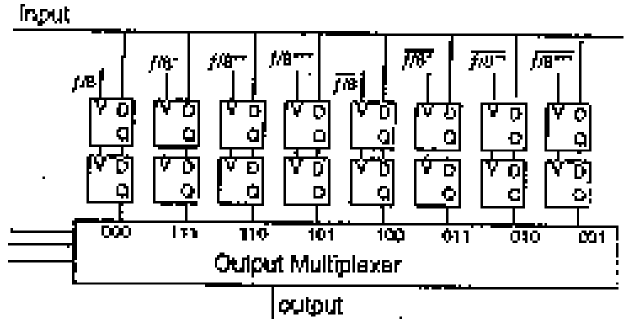

Figure 15. Eight-Parallelized 16-bit Shift Register

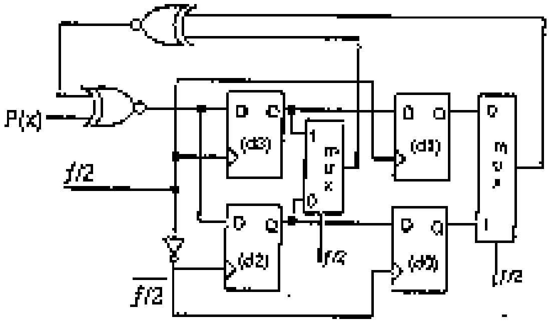

Figure 17. Linear Feed-back Shift Reg.

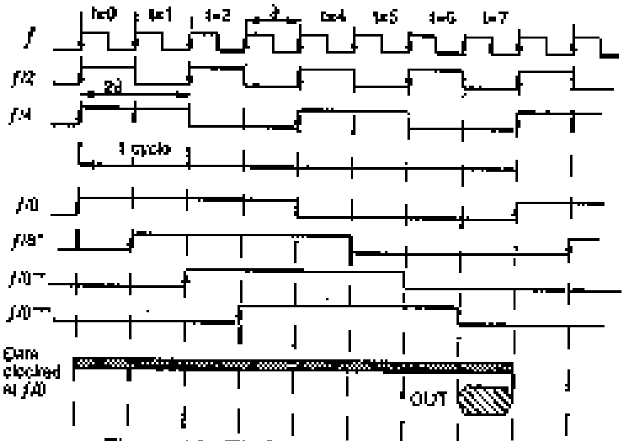

Figure 16. Timing Diagram

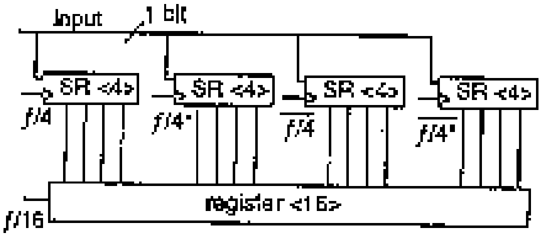

Figure 18. Parallel-serial Convertor

according to $P = f/2 * C' * Vdd^2$, with $C' < C$. The basic frequency is reduced of a factor 2 and the switched capacitance C' can be reduced with smaller transistors for the D flip-flops, at the exception of the last one and of the output multiplexer. The speed is only given by the last flip-flop and by the output multiplexer, while the first flip-flop has to pick up the data at the frequency *f*. The switched capacitance of the clock node is also significantly reduced. Figure 15 shows an 8-parallelized shift register and Figure 16 the corresponding timing diagram.

For shift registers that are not the critical path at a given low Vdd, the design of a parallelized slower shift register is acceptable. The degree of parallelism can be increased (Fig. 15) with a more complex clock generator (Fig. 16), resulting in more power savings. However, the delay of the output multiplexer slows down the shift register compared to a non parallelized structure at the same Vdd.

Shift register parallelization can be used for linear feed-back shift registers [11] with as many output multiplexers as the number of inputs of the XOR tree. Figure 17 shows an example with two output multiplexers. The example in [11] is a M-parallelized M-bit shift register with M-input simplified multiplexers.

A parallel-serial converter can be designed with a load M-parallelized shift register. The N-bit word is loaded in the shift register and clocked at *f*/M to the 1-bit output through the output multiplexer. Figure 18 shows the parallelized structure of a 16-bit serial-parallel converter in which the 1-bit input is successively loaded in four 4-bit shift registers clocked at *f*/4. Power consumption is reduced of a factor 4 with the same throughput. As there is no output multiplexer, the maximum frequency of such a structure can be much higher than the non parallelized serial-parallel converter. Reference [12] describes another architecture with a cross-access memory dedicated for a multi-way serial-parallel converter.

A classic D-Flip-Flop, implemented with two latches in series, can be parallelized while using two latches in parallel with an output multiplexer. Such a flip-flop, already provided by some libraries [3, 7, 10], is sensitive to both edges of the input clock signal *f*/2 [13]. Its use in synchronous systems results in a master clock reduced of a factor 2 [7]. Any finite state machine may be implemented with double edge D-flip-flops to reduce the input frequency by a factor 2 and the power consumption of the clock tree.

8. Conclusion

Logic design at low supply voltage has to be performed while considering the critical path. Logic modules on the critical path have to be decomposed and parallelized in a different way than those which are not on the critical path. Parallelization of logic blocks is a very effective technique to reduce the power consumption. Examples of parallelized synchronous counters show that the power consumption can be saved while reducing the supply voltage. Parallel shift registers present a reduced power consumption at the same Vdd.

References

[1] V. von Kaenel et al, "Automatic Adjustment of Threshold & Supply Voltage Minimum Power Consumption in CMOS Digital Circuits", 1994 IEEE Symposium on Low Power Electronics, San Diego, October 10-12, 1994, pp. 78-79.

[2] D. Liu, C. Svensson, "Trading Speed for Low Power by Choice of Supply and Threshold Voltages", IEEE JSSC-28, No 1, Jan 1993, pp. 10.

[3] C. Piguet et al. "Low-Power Low-Voltage Digital CMOS Cell Design", Proc. PATMOS'94, Oct. 17-19, 1994 Barcelona, Spain, pp. 132-139.

[4] A. P. Chandrakasan, S. Sheng, R. W. Brodersen, "Low-Power CMOS Digital Design" IEEE JSSC, Vol. 27, No 4, April 1992, pp. 473-484.

[5] R. F. Lyon, "Cost, Power, and Parallelism in Speech Signal Processing" , IEEE 1993 CICC, Paper 15.1.1, San Diego, CA, USA.

[6] E. A. Vittoz, "Low Power Design : Ways to Approach the Limits", Plenary Address ISSCC'94, February 16-18, 1994, San Francisco, USA.

[7] J-M. Masgonty et al, "Technology and Power Supply Independent Cell Library" IEEE CICC'91, May 12-15, 1991, San Diego, CA, USA, Conf. 25.5

[8] F. Moraes, N. Azemard, M. Robert, D. Auvergne, "Flexible Macrocell Layout Generator", 4th ACM/SIGDA Physical Design Workshop, Los Angeles, 1993, pp. 105-116.

[9] C. Mead, M. Rem, "Cost and Performance of VLSI Computing Structures" IEEE JSSC-14, April 1979, pp. 455-462.

[10] C. Piguet, "Ultra Low-Power Digital Design", Low-Power/Low-Voltage IC Design Course, May 9-13, 1994, Monterey, CA, USA, and April 17-21, 1995, Santa Clara, CA, USA

[11] M. Lowy, "Low-Power Spread Spectrum Code Generator Based on Parallel Shift Register Implementation", 1994 IEEE Symposium on Low Power Electronics, San Diego, October 10-12, 1994, pp. 22-23.

[12] Y. Ohtomo, M. Suzuki, "A 250-Mb/s, 700-mW, 32-Highway *8-b S/P Converter LSI with Cross-Access Memory", IEEE JSSC-27, No 4, April 1992, pp. 530-38.

[13] R. Hossain et al. "Low-Power Design Using Double Edge Triggered Flip-Flops", IEEE Trans. on Very Large Scale Integr. Syst. Vol. 2, No 2, June 1994, pp. 261.

[14] J.-P. Hayes, "Computer Architecture and Organization"; McGraw-Hill, 1978, p. 382

[15] G. Panigrahi, "The Implications of Electronic Serial Memories", Computer, July 1977, pp. 18-25