

Differential and Pass-Transistor CMOS Logic for High-Performance Systems

Vojin G. Oklobdžija, Fellow IEEE

Abstract - This paper presents a review of differential and pass-transistor logic used in today's high-performance systems. Various circuit and logic design styles used in contemporary high-performance processors have been reviewed. The new logic is advantageous over standard CMOS in terms of performance and very often in terms of: area, speed and power as well. Evolution of various high-performance latches has been presented.

I. INTRODUCTION

Computational and market demands have driven VLSI microprocessors into doubling of their performance every three years as shown in Fig.1. In 1994 the first microprocessor (known under the code name of "Alpha" from Digital Equipment Corporation) delivered a single-chip performance equivalent to that of the CRAY-1 supercomputer [12]. However, since its introduction in 1993, the performance of the "Alpha" processor has tripled delivering 40 SpecInt95, as reported in 1997 [13]. Similarly the same trend is observed in the "mainstream" computer market represented by the X'86 architecture. The clock frequencies have reached 600MHz [13] and are expected to top 1GHz in the next year. This demand has had its repercussions on the circuit techniques and the design style used to design high-performance systems.

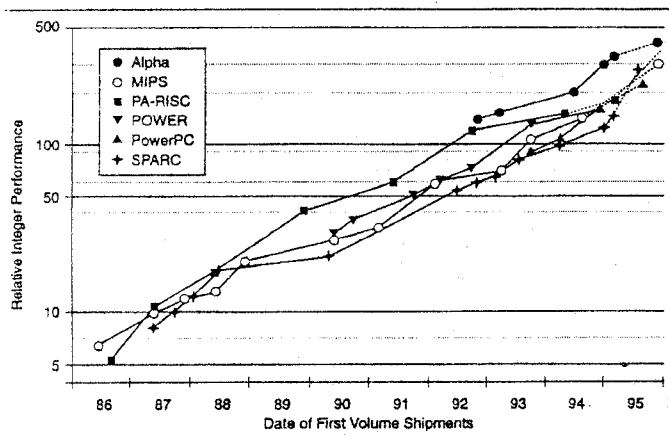


Fig. 1. Performance increase in RISC microprocessors [11]

The author is with Electrical and Computer Engineering,
University of California, Davis, CA 95616;
E-mail: voj@ece.ucdavis.edu

Keeping this rate of performance increase is not possible only through the advances in fabrication technology. Therefore the improvements in all the other aspects of the design are necessary to support the rate of this progress. Importance of a good circuit design became apparent with the recent introduction of the third generation of "Alpha" processor 21264 which performance surpassed all the other processors introduced this year by a wide margin [13].

As the technology reaches into the deep sub-micron region, the use of regular CMOS came to its limits. The problems associated with the power and speed required that the other types of logic family be examined. In order to reach the performance goals, it is not uncommon to see the use of dynamic logic in the critical paths of a processor. Quite the contrary, almost every high-performance processor today uses some of the non-conventional CMOS design techniques such as: *Domino* logic (single ended or differential) [14,1] as well as *pass-transistor* design techniques. The circuit implementation of the critical part of a high-performance processor is so important that it is essential for the leading processor design centers today to have a very good circuit design team. The interaction between the circuit and the architecture group became so close that it has almost eliminated logic design or confined it to a very small portion of a non-critical parts of the processor [13].

The use of pass-transistors regained interest in the institutions possessing the state of the art technology. This design style was re-examined and it yielded impressive results. It was not only shown that the substantial performance gains can be achieved over the conventional design style, but that the power-delay product of such logic was lower. The power has increasingly becoming an issue of importance as the processor has been migrating into the consumer market, especially portable and hand-held devices.

II. DIFFERENTIAL LOGIC

The introduction of differential CMOS logic evolved from the development of dynamic CMOS such as "Domino Logic" [14] and exploration of the circuit families that are

to replace nMOS logic in the early 1980s. This development took place within IBM and AT&T Bell Laboratories and resulted in several new circuit and logic configurations.

A. CVS logic

Cascode Voltage Switch Logic (CVSL) was developed in IBM [1] as an improvement over the use of pseudo n-MOS. It comes in two forms: single-output and differential output (or double-rail). The later form of the logic is also called DCVSL (Differential Cascode Voltage Switch Logic).

DCVSL is made of two n-type switching networks, one implementing \bar{f} and the other f , and of two p-type transistors, connected in a cross-coupled combination to V_{dd} , used as pull-up devices (Fig.2). Depending on the state of the differential inputs, either node N1 or N2 is pulled down by one of the nMOS logic tree (but never both). The regenerative action of the pMOS latches keeps the outputs Q and \bar{Q} static and assures the full voltage swing, V_{dd} or ground, of its outputs.

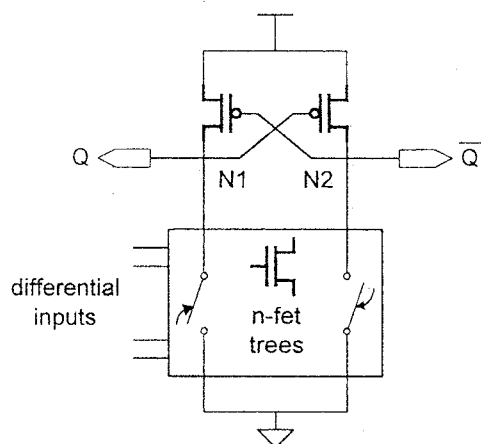


Fig. 2. Static DCVSL logic

The two logic trees are capable of processing complex functions within a single circuit delay. A tree with N n-type devices is capable of computing a function with up to $(2^N - 1)$ input variables.

The advantage of DCVSL logic is that both polarities of the output are represented, thus inversion operation is not necessary. This eliminates the need for the inverter and makes this type of logic inherently faster. The presence of both polarities of the output has other advantages as well. If the circuit is operating correctly, the values of the output signals can only assume 0-1 or 1-0, i.e. the 0-0 or 1-1 combination can never occur. This gives this logic "self-checking" properties. If one of the forbidden combinations is detected, it is immediately signaled as a failure of the logic and the system switches to the appropriate action.

Another variations of CVSL are Static and Dynamic CVSL circuit. Dynamic logic is available in two forms:

single-ended (single-output) and double-ended (where true and complements of the function are present).

One the problem of static DCVSL is the signal asymmetry which can appear during the transition. Given that the pMOS transistors are the only pull-up devices there may be a time window during which both the pMOS and the nMOS are ON. This situation will create a current from V_{dd} to ground node causing current spikes and additional delay. The choice of the size of the pMOS is thus very important. If the pMOS is made too small the transition of the signal from GND to V_{dd} is too slow. If on the other hand, the pMOS devices are made too big the transition of the output node from V_{dd} to ground is too slow. This makes static CVS to be a "ratioed" logic. In general to assure a good "pull-up" of the output signal the pMOS devices should be twice the size of the nMOS devices. There is no direct current from V_{dd} to ground after the transition occurred, however because of the asymmetry of the circuit the power consumption of CVSL.

B. CVSL versus CMOS

The main difference between CMOS and DCVSL is in the way the switching function is implemented. While both CMOS and DCVSL implement the true function and its complement, DCVSL uses only n-type devices for both switching trees whereas CMOS use p-type for the f tree and n-type for \bar{f} tree.

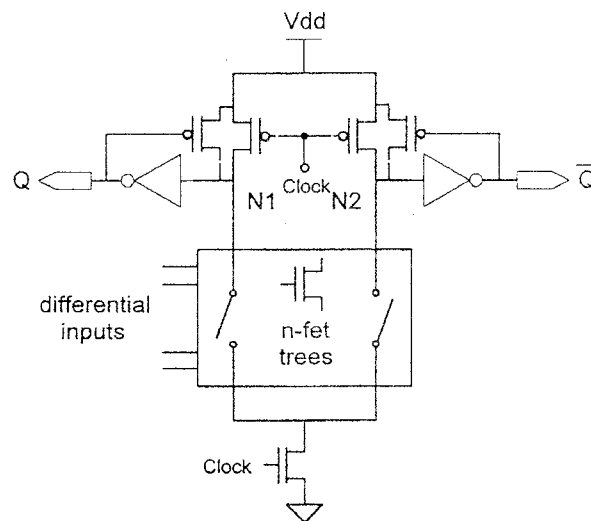


Fig. 3. Dynamic DCVSL logic

In terms of area this allow CVSL to be smaller than CMOS. Since the carrier mobility in the pMOS transistor is half of that in the nMOS, the pMOS transistors need to be made twice as large. Therefore the area taken to implement the switching tree representing the function f is usually twice as large as compared to the switching tree representing \bar{f} . In DCVSL those switching trees are approximately the same. In addition to the area reduction,

the use of nMOS transistors results in a reduced input capacitance thus contributing to the speed of the circuit.

In general, due to the lower input capacitance and a better intrinsic transistor speed CVSL should be faster as compared to CMOS using the same transistor sizes. In studies done by IBM, CVSL has shown an overall performance improvement [1]. Other studies [2], show an improvement of performance but at the cost of increased power consumption.

In terms of the number of transistors, CVSL uses two extra pMOS transistors in the cross-coupled combination, as compared to CMOS. However the implementation of both functions f and \bar{f} doesn't necessarily mean that duplication of the transistors is necessary. A number of transistors can be shared between f and \bar{f} switching trees. The amount of such overlap is dependent on the function. Thus the number of transistors in CVSL is generally the same or lower as compared to CMOS.

The sharing of the transistors is illustrated in the example of a 3 input XOR gate shown in Fig.4.

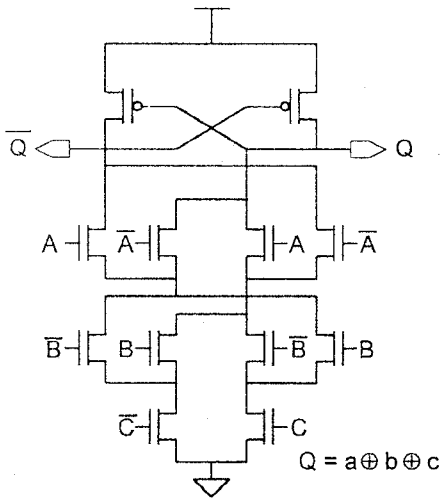


Fig. 4. 3-Input XOR implementation in CVSL

III. PASS-TRANSISTOR LOGIC

New CMOS logic families using pass-transistor circuits have recently been proposed with the objective of improving speed and power [4,6]. Two of them, simultaneously developed by Hitachi: CPL [4] and DPL [6], are the most notable. The Double Pass-Transistor Logic, developed by Hitachi in 1993 demonstrated an 1.5nS 32-bit ALU in 0.25 μ m CMOS technology [4] and a 4.4nS 54X54 bit multiplier [9]. New developments followed from IBM and from Toshiba introducing DCVSL-PG [3] and SRPL [5]. Recent studies have shown that the use of pass-transistor logic not only brings speed and area improvement, but also results in lower power.

A. CPL

In 1990, researchers from Hitachi Central Research Laboratories in Japan published the structure known as Complementary Pass-Transistor Logic (CPL) [4]. The CPL was significant in the fact that it was based on the use of the pass-transistor networks. The logic function, which is built from the pass-transistors, not only efficiently utilizes the silicon, but results in a very fast logic which is also characterized by low-power consumption.

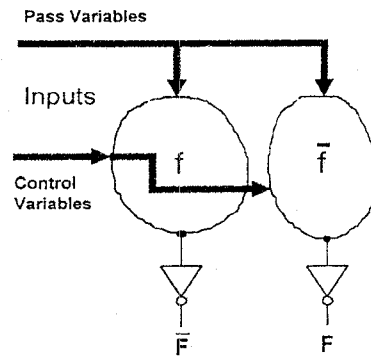


Fig.5. CPL logic structure

The general structure of CPL is shown in Fig.5. The given function f is implemented from two pass transistor logic blocks implementing the function f and its complement \bar{f} . Such obtained logic is differential as every variable is represented in its *true* and *complement* form.

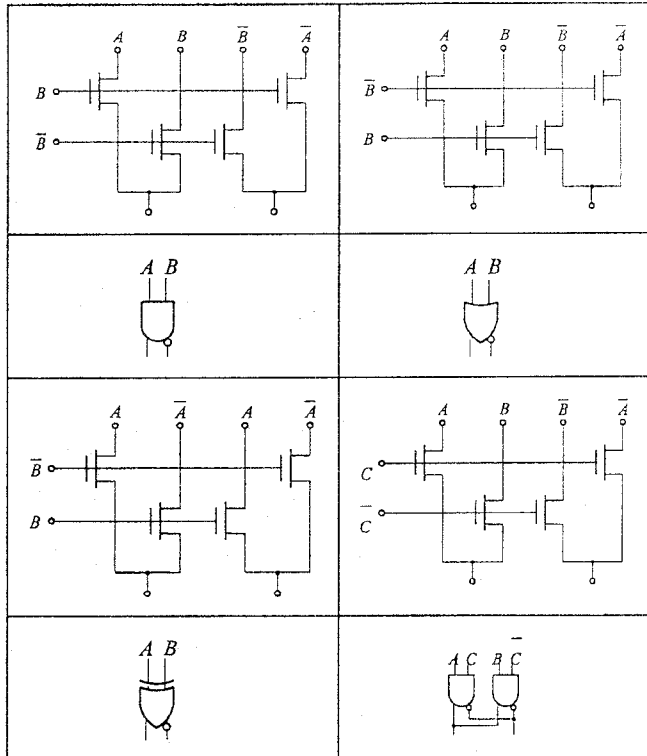


Fig.6. CPL circuit implementation of basic logic functions

If we are to implement an AND gate, a NAND output will be readily available. Therefore, complementation consists of a proper choice of the signals only, given that both polarities are available. The CPL basic gates are shown in Fig. 6.

A family of gates is implemented in this fashion including the XOR / XNOR combination as well as multiplexer. A distinguished feature of CPL circuits is that the implementation of the multiplexer circuit is especially effective and fast. The same circuit topology is used to implement an XOR gate resulting in equally fast and efficient realization. This feature has much importance in digital system design given that multiplexer and XOR gates are essential building blocks which are found in the critical paths of various components. A CPL implementation of and XOR gate and sum bit of full-adder are shown in Fig.7.

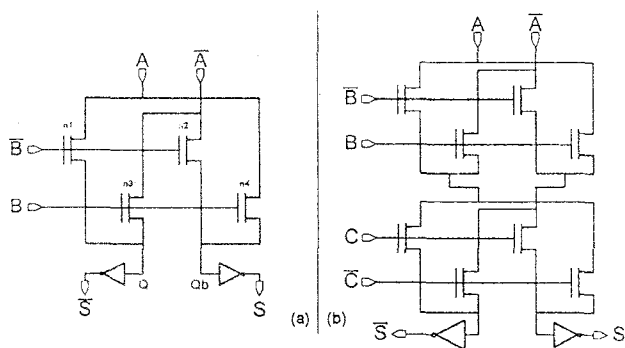


Fig.7. Basic CPL gates: (a) XOR (b) Sum circuit

CPL proved to be not only very efficient but also very fast, yielding an 3.8nS 16X16b multiplier in double metal 0.5μ CMOS technology [4]. However, CPL suffered from the problem of signal degradation. When passed through a series of pass-transistors, the signal voltage is degraded by one V_T (threshold drop). This brings the transistor in the inverter to the conducting region, causing static current to flow from V_{CC} to GND resulting in an increase in static power. To alleviate this problem, Hitachi researchers used two types of transistors: logic transistors (with $V_T = 0V$) and transistors used in the inverter (with $V_T = 0.4V$ and $-0.4V$). Though this reduced static power dissipation and delay time, it increased the process complexity and the sensitivity to noise. In the new version of CPL [8], the problem of the "threshold drop" was alleviated by using a special type of inverter which has the ability to restore the voltage level to its full potential. This inverter is shown in Fig.8. The distinguishing feature of this inverter is that the feedback which brings the input to the full voltage swing (eliminating the V_T drop) is independent of the output load of the inverter. A fast restoration of the full signal swing is possible thus minimizing the power consumed during this transition.

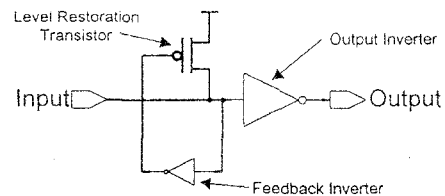


Fig.8. CPL inverter

A clever use of fast feedback is used in this special type of CPL inverter. The restoration of the signal level is independent on the load at the output, thus resulting in faster signal level restoration and decrease in power during the signal transition.

The concept of CPL has been further extended into a design style associated with the tool for automatic generation of the logic block named "Lean Integration" [8]. The use of this design style has provided a beyond marginal improvements in performance, power and area of the ASIC and micro-processor units. Another advancement of the CPL concept termed LEAP has been reported recently [15].

B. DPL

A pass-transistor logic attempts to solve the problem of the pass transistor threshold voltage drop exhibited in CPL. DPL evolved from the same group of researchers at Hitachi Central Research Laboratories lead by Okhubo [6]. The logic is named DPL for Double Pass-transistor Logic. DPL therefore represents a "pass-transistor logic" family alternative to CPL. In creating the switching network f , DPL uses both: n-MOS and p-MOS transistors in parallel. This eliminates the problem of the "threshold drop" and the use of inverters after each logic block. Elimination of inverters results in enhanced speed, however, buffering of the signal after every 2-3 stages is necessary.

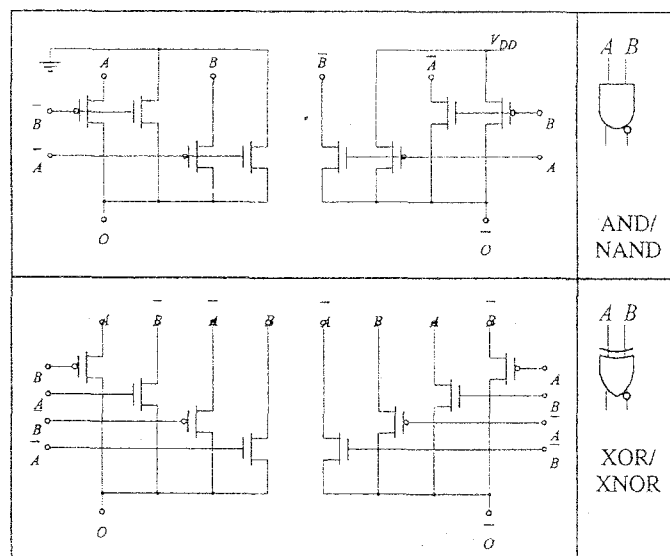


Fig.9. DPL circuit implementation of basic logic functions

The two basic gates used in DPL are shown in Fig.9. The simplicity of DPL family is apparent. For this logic family to be complete it is necessary to implement only on logic function (AND/NAND) and inversion which is obtained by simply choosing an appropriate output. To achieve an efficient implementation of XOR gate is also necessary. As in CPL the basic circuit structure in DPL is a multiplexer which topology is equivalent to that of an XOR gate. However, unlike in CPL those two basic building blocks (XOR and MUX) do not necessarily have to be followed by an inverter, thus making an implementation of a pass-transistor chain possible. When the signal is propagated through several stages of pass-transistors, restoration of the signal is necessary which is achieved by inserting inverters. Unlike in CPL it is not necessary for this inverter to be of a special kind.

Hitachi has shown two very fast implementations using DPL: one a 1.5nS 32-b ALU [6] and another a 4.4nS 54X54-b parallel multiplier [9]. An XOR and Sum bit of a full-adder are shown in Fig. 10.

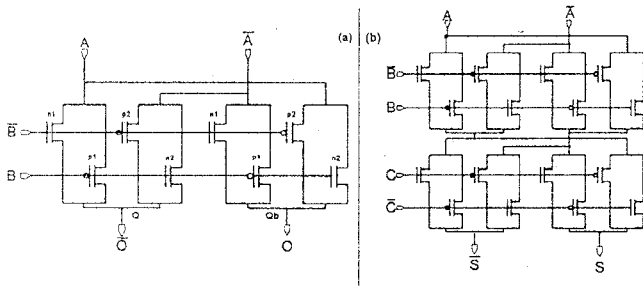


Fig.10. DPL Logic: (a) XOR (b) One bit full-adder: Sum circuit

B. DVL

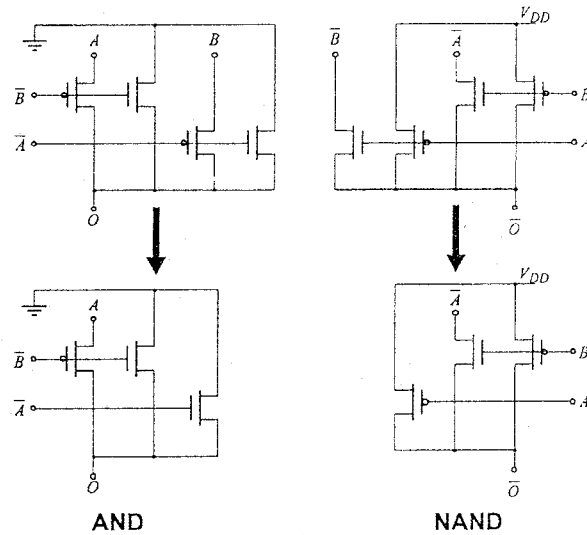
A step further in development of DPL is taken in a logic family termed DVL (Dual Value Logic) [10]. The new logic family was obtained from DPL by elimination of the redundant branches and rearrangement of signals. These simplifications still preserve full swing operation of DPL and improve its speed. The speed improvement is a direct result of elimination of one branch containing one transistor. This minimizes the capacitive load "seen" by the previous gate by minimizing the number of inputs and number of capacitive loads.

The new logic family is achieved in three steps:

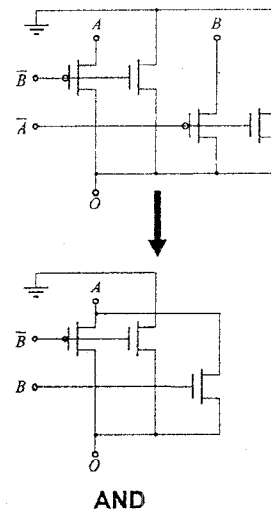
- (a) elimination of redundant branches in DPL
- (b) elimination of branches via signal rearrangement
- (c) combination of (a) and (b) using two faster halves

The process is illustrated in Fig.11.(a),(b),(c)

A faster half was chosen from (a) and from (b) resulting in a complete gate (c). Fortunately (a) produces a faster NAND while (b) produces a faster AND, which makes a complete gate shown in Fig.11.(c).



(a) Elimination of redundant branches



(b) Signal re-arrangement

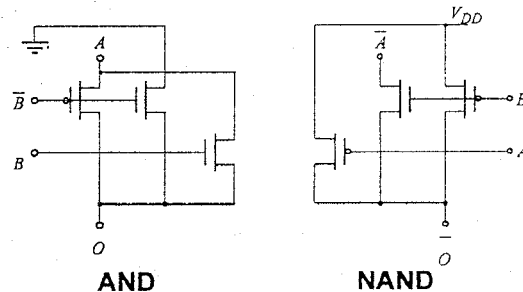


Fig.11. DVL Logic: (c) Resulting DVL gate is obtained by taking two faster halves from (a) and (b)

The resulting DVL gate contains total of 8 transistors (3 p-transistors and 3 n-transistors) compared to 4 transistors of each type in DPL. There is a total of 9 inputs in DVL

versus 12 in DPL resulting in a smaller capacitive load of DVL gates. Of those inputs 3 are connected to the transistor source and 6 to the gate: 3 to p-type and 3 to n-type. In DPL 8 inputs are connected to the source 4 to p-type and 4 to n-type transistors. The total area (taking re-sizing into account) is only 5% larger in DVL gate. The speed advantage is 20% in favor of DVL.

The comparison between NAND/AND DPL gate and NAND/AND DVL shows:

- 20% speed improvement, utilizing 75% of the transistors used in DPL.
- 25% less connections and wires as compared to a DPL gate.
- The 4% area increase in comparison to DPL is not found to be substantial.

A similar method is used to build the NOR/OR gates.

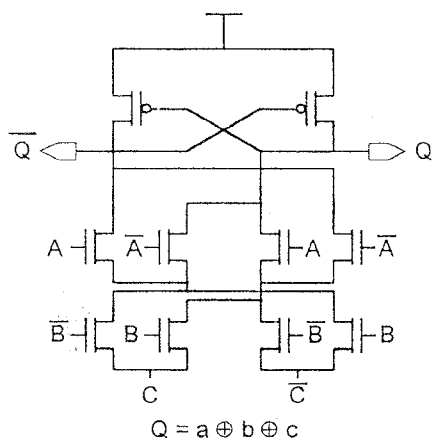


Fig.12. 3-input XOR Gate implementation in DCVSL-PG

C. DCVSL-PG

Further development of differential CMOS family is presented in the paper by Lai and Hwang [3]. They introduced pass-transistor logic in the DCVS logic tree in order to eliminate the problem of current spikes. They have solved this problem by having the switching tree act as the pull-up (accelerating the shut down of the p-transistors). The cross-coupled pMOS is acting as a load to regenerate the output signal level (Fig.12). The size of the pMOS transistors is not critical anymore. They can be made of the minimal size, thus unlike DCVSL, DCVSL-PG is not a ratioed logic. In addition there is fewer transistors in DCVSL-PG leading to a smaller and faster circuits compared to DCVS. The main difference compared to DCVSL is in the logic nMOS trees. In DCVSL-PG they are not always connected to ground but are, most of the time, connected to pass variables or, sometimes, to supply voltage. The switching network thus does not act as a path to ground but also passes the input variables to the output. The cross-coupled pMOS pair is only used as a regenerative load to bring the outputs to full-swing level.

DCVSL-PG logic showed a performance better than that of DCVS. This was demonstrated by an implementation of 2nS 64-bit adder in 0.5μ CMOS technology.

C. CVSL-PG

Researchers from Toshiba Corp. developed their version of differential CMOS pass-transistor logic that does not suffer from degraded pull down performance [5]. They named it Swing Restored Pass-Transistor Logic (SRPL). In SRPL the generic gate consists of a pass-transistor logic constructed of nMOS transistors (similar to CPL) and a latch type swing restoring circuit consisting of two cross-coupled CMOS inverters (Fig.13.). The nMOS transistor logic network implements any Boolean logic function while the complementary outputs of the pass-transistor logic are restored to full swing by the cross-coupled combination at the circuit output. In this way SRPL solves a major problem of the CPL logic. However, it is argued that the input variable can "see" a long chain through several gates, thus making the total output capacitance of the circuit quite large. Toshiba has built an experimental MAC (Multiply Accumulator) in a 0.4μ CMOS technology achieving a 150MHz speed at 3.3V supply voltage.

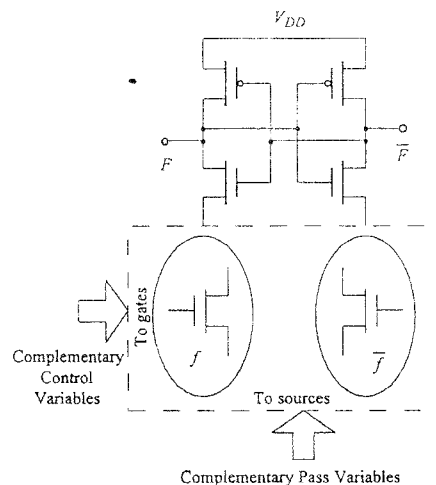


Fig.13. Generic SRPL Gate

Comparisons of full adder circuits implemented with CMOS, CPL, DPL, DCVSPG and SRPL showed CPL to be the fastest followed by SRPL and DCVSPG logic. However, SRPL had the best power-delay product which amounted to 21% of that of CMOS [5].

IV. LATCHES

An important part of every high-performance system is the latch. At the increasing clock frequencies very little time is left for computation. The overall speed of those systems is enhanced by deep pipelining and the use of relatively small number of logic stages. The fact that the delays associated with wires, clock-skew and the jitter

introduced by the PLL are not scaling with technology makes this situation even worse. Therefore an increasing demand has been placed on the latch requiring to minimize the amount of time which is not contributing to the computational cycle such as: the latch setup time and latch delay. Several new and unusual latch configuration have emerged in recent high-performance processors.

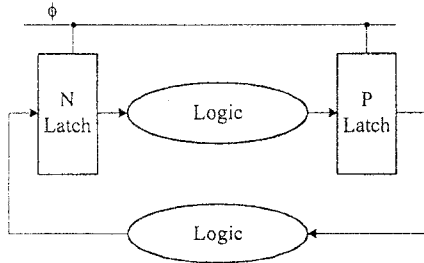


Fig. 14. Single pipeline stage utilizing both polarities of the clock

The diagram of single-phase clocked pipelined system, consisting of two logic blocks separated by N and P type latches is shown in Fig. 14. N type latches are transparent when Clock = 1, and opaque when Clock = 0, while P type latches are transparent when Clock = 0, and opaque when Clock = 1. Since the pipeline design is based on latches, they play the key role in overall system performance.

A. TSPC-Latch

TSPC technique is commonly used in high performance digital systems due to its simplicity and fast operation [16]. Four basic stages exist in TSPC, pre-charged N and P, and non-precharged N and P, as shown on Fig.15. By combining these stages latches and flip-flops can be formed. For example, N type latch consists of two non-pre-charged N stages (Fig.16).

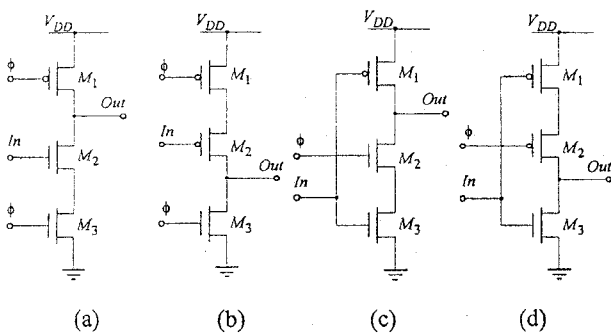


Fig. 15. Basic CMOS TSPC stages: a) pre-charged N, b) pre-charged P, c) non-precharged N, d) non-precharged

B. "Alpha"-Latch

A typical example of a demand on a latch in a high performance processor is the evolution of the latch used in Digital "Alpha" processor. The first generation of "Alpha" 21064 [12] used modification of TSPC latch (Fig. 17.) the modification over TSPS latch is in additional transistor

added to eliminate floating nodes and improve the immunity to noise of this latch.

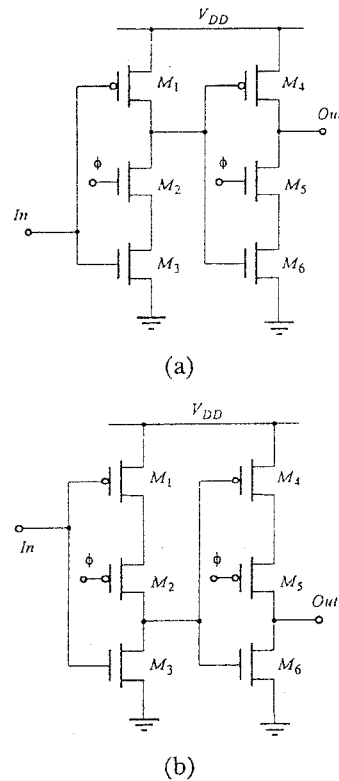


Fig. 16. TSPC latch (a) N-type (b) P-type

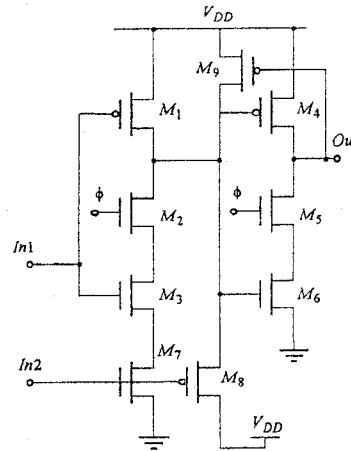


Fig.17. Modified TSPC latch as used in 21064, the first generation "Alpha" processor from Digital [12]

In the second generation "Alpha" processor 21164, Digital designers have opted for a very shallow latch its main part consisting of the pass-transistor switch in order to reach 300MHz operation [17]. The modification of this latch consist of introducing a logic gate at the input, thus being able to perform a logic NAND operation.

Demand for even higher clock rate of 600MHz had its effect on the latch design. The third generation "Alpha", 21264 uses a differential latch resembling a sense amplifier

in a memory cell. The propagation delay of this latch is 450nS [13].

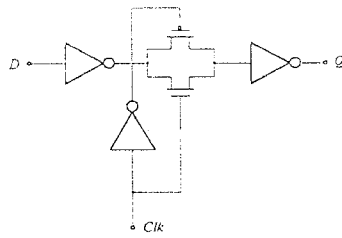


Fig.18. The latch used in the second generation "Alpha" processor from Digital 21164 [17]

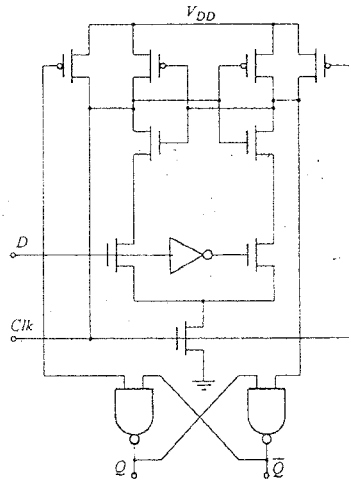


Fig.19. The latch used in the third generation "Alpha" processor from Digital 21264. The latch is differential [13]

V. CONCLUSION

In this paper various circuit and logic design styles used in high-performance processors have been reviewed. The new logic has advantages over standard CMOS in terms of performance and very often in terms of: area, speed and power as well. A very important aspect of a high-performance system is the clocking methodology and associated latch design. Evolution of various high-performance latches has been presented.

REFERENCES

[1] LG Heller, WR Griffin, et al, "Cascode Voltage Switch Logic : A Differential CMOS Logic Family" 1984 *IEEE International Solid-State Circuits Conference*, vol 27, pp16-17 February 1984.

[2] KM Chu and DL Pulfrey, "A Comparison of CMOS Circuit Techniques : Differential Cascode Voltage Switch logic Versus Conventional Logic" *IEEE J.Solid State Circuits*, vol sc 22, pp528-532, 1987.

[3] F.S. Lai and Hwang, "Differential Cascode Voltage Switch with Pass Gate Logic Tree for High Performance CMOS Digital Systems", 1993 *International Symposium on VLSI Technology, Systems and Applications*, pp358-362, May 1993.

[4] Yano, K, et al, "A 3.8 ns CMOS 16 *16 b Multiplier Using Complementary Pass-Transistor Logic", *IEEE J. Solid State Circuits*, vol 25, p388-395, April 1990.

[5] Akilesh Parameswar, et al, "A Swing Restored Pass-Transistor Logic Based Multiply and Accumulate Circuit for Multimedia Applications", *IEEE 1994 Custom Integrated Circuit Conference*, pp278-281.

[6] Makoto Suzuki, et al, "A 1.5 ns 32 b CMOS ALU in Double Pass-Transistor Logic", 1993 *ISSCC Dig. Tech. Papers*, pp90-91, February 1993.

[7] KM Chu and DL Pulfrey, "Design Procedures for Differential Cascode Voltage Switch Circuits", *IEEE J.Solid State Circuits*, vol sc 21, no 6, December 1986.

[8] Yano, K, et al, "Lean Integration : Achieving a Quantum Leap in Performance and Cost of Logic LSIs", *Proceedings of the IEEE 1994 Custom Integrated Circuit Conference*, May 1-4, 1994, San Diego, California, p.603-606.

[9] N. Ohkubo, et al, "A 4.4nS CMOS 54x54-b Multiplier Using Pass-Transistor Multiplexer", *Proceedings of the IEEE 1994 Custom Integrated Circuit Conference*, May 1-4, 1994, San Diego, California, p.599-602.

[10] V.G. Oklobdzija, B. Duchene, "Pass-Transistor Dual Value Logic For Low-Power CMOS," *Proceedings of the 1995 International Symposium on VLSI Technology*, Taipei, Taiwan, May 31-June 2nd, 1995.

[11] L. Gwennap, "Processor Performance Climbs Steadily", *Microprocessor Report*, p.18, January 23, 1995.

[12] Dobberpuhl, D., et al, "A 200 MHz 64 b Dual-Issue CMOS Microprocessor," 1992 *IEEE International Solid-State Circuits Conference, Digest of Technical Papers*, San Francisco, CA, USA, 19-21 Feb. 1992, pp. 106-7, 256.

[13] B. Gieske, et al, "A 600MHz Superscalar RISC Microprocessor with Out-of-Order Execution", 1997 *ISSCC Dig. Tech. Papers*, p.176-177, February 7, 1997.

[14] R. Krambek, et al, "High-Speed Compact Circuits with CMOS", *IEEE Journal of Solid-State Circuits*, Vol.SC-13, No.3, June 1982.

[15] K. Yano, et al, "Top-down pass-transistor logic design", *IEEE Journal of Solid-State Circuits*, Vol.31, p.792-803, June 1996.

[16] Y. Ji-Ren, I. Karlsson, C. Svensson, "A True Single-Phase-Clock Dynamic CMOS Circuit Technique," *IEEE JSSC*, vol. SC-22, 1987, pp. 261-266.

[17] B. Benschneider, et al, "A 300MHz 64-b Quad-Issue CMOS RISC Microprocessor", *IEEE Journal of Solid-State Circuits*, Vol.30, No.11, November 1995.